AD-785 360

COMMUNITY NOISE EXPOSURE RESULTING FROM AIRCRAFT OPERATIONS: COMPUTER PROGRAM OPERATOR'S MANUAL

Nicolaas H. Reddingius

Bolt Beranek and Newman, Incorporated

Prepared for:

Aerospace Medical Research Laboratory

July 1974

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Organizations and individuals receiving announcements or reports via the Aerospace Medical Research Laboratory automatic mailing lists should submit the addressograph plate stamp on the report envelope or refer to the code number when corresponding about change of address or cancellation.

Do not return this copy. Retain or destroy.

Please do not request copies of this report from Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22151

This report has been reviewed and cleared for open publication and/or public release by the appropriate Office of Information (OI) in accordance with AFR 190-17 and DODD 5230.0. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Henning E. von Gierke

HENNING E. VON GIERKE
Director
Biodynamics and Bionics Division
Aerospace Medical Research Laboratory

AIR FORCE/56780/3 September 1974 — 100

ic

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AMRL-TR-73-108	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD 785 360
4. TITLE (and Subtitle) COMMUNITY NOISE EXPOSURE RESULTING FROM AIRCRAFT OPERATIONS: COMPUTER PROGRAM OPERATOR'S MANUAL		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER 2582
7. AUTHOR(s) Nicolaas H. Reddingius		8. CONTRACT OR GRANT NUMBER(s) F-33615-73-C-4160
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 21120 Vanowen Street (P.O. Box 633) (91305) Canoga Park, California 91303		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62202F 72310424
11. CONTROLLING OFFICE NAME AND ADDRESS Aerospace Medical Research Laboratory, Aerospace Medical Division, Air Force Systems Command, Wright-Patterson AFB, Ohio 45433		12. REPORT DATE July 1974
		13. NUMBER OF PAGES 235
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

aircraft noise
airport planning
noise exposure forecast
community noise exposure

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A user oriented description of a computer program to calculate community noise exposure due to aircraft operations is given. Formal definition of all allowable card sequences and examples of coding for all types of aircraft operations are presented as well as guidelines for efficient use. The program which is entirely written in FORTRAN IV produces printed output as well as output compatible with the CALCOMP GPCP contouring package. A discussion of the architecture of the program and the interpretation of the output can be found in companion volumes AMRL-TR-109 and AMRL-TR-73-105, respectively.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

PREFACE

This report is one of a series describing the contractual and in-house research program undertaken by the Aerospace Medical Research Laboratory under Project/Task 723104, Measurement of Noise and Vibration Environments of Air Force Operations, to develop a procedure for predicting the community noise exposure resulting from aircraft operations. The companion reports are listed as references 1, 2, 3, 4, and 5. The Air Force Weapons Laboratory provided funding to support development of this program.

This report describes a computer program to calculate the community noise exposure as determined by the Noise Exposure Forecast (NEF) methodology. Studies by the United States Environmental Protection Agency have meanwhile led to the specification of another community noise exposure measure: the Day/Night Average Level (DNL). Since the two types of descriptors are essentially similar, the computer program may be adapted to this new unit when the need arises without the need to redevelop the computational algorithms.

The author wishes to acknowledge the assistance of Richard D. Horonjeff in developing the mathematical model for flight operations implemented in the program and in preparing Section IV of this manual. R. Rao Kandukuri's help and perseverance in debugging and checking out the program was indespensible. Myles A. Simpson provided valuable editorial comments from the point of view of a user only moderately familiar with data processing.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

LIST OF TABLES

SECTION I

INTRODUCTION TO THE NEFUSAF PROGRAM

This manual describes a set of FORTRAN programs which make up the NEFUSAF package. The purpose of this package is to calculate Noise Exposure Forecast (NEF) values around military air bases.

The purpose of this manual is to teach the user how to use the NEFUSAF package and how to avail himself of its many features. The programs were written in the FORTRAN IV language in conformity to USAF Standard R&D-28-AFWL (AFWL Computer Programming Standards). Further programmer oriented documentation as well as a technical discussion of the Noise Exposure Forecast concept and its computer implementation may be found in the other volumes of this technical report.

It has been attempted to write this manual for a user with a minimal knowledge of computing. In order to successfully communicate the special considerations of output other than on the line printer, more computer knowledge on the part of the user had to be assumed in Section II than in the rest of the manual. In order to make Figure II.1 accessible to users unschooled in language theory, we have included a short discussion of BNF in Section II.

Basic input to the program is via punched cards. All cards have an alphabetic code in columns 1-6. The code is always left-justified in this field. The possibility exists that all required data of a particular type does not fit onto one card. In that case a nonblank character should be placed in the field of columns 79-80, and one may continue on the next card. This continuation card is the only one which has no entry in its first 6 columns.

Input cards which are recognizable by the program fall in three categories. The first type of card regulates the operation of the program and performs on request certain control functions; these are the control cards. There are two types of data cards, sequence dependent and sequence independent. The sequence dependen cards describe the aircraft operations. Sequence dependency as defined here implies that sequences of this type of card must be in a particular order, irrespective of the presence of other types of cards which may be interspersed. The sequence independent data cards contain noise and performance data of the aircraft. They are sequence independent in that they can appear anywhere without destroying the sequence of sequence-dependent cards in between which they are inserted. There is, of course, still a logical sequence dependence for these cards, since the information on them must be available to the computer at the time it is needed.

Since the input to the program is very complex and the execution of the program quite time consuming, the program may be run in two modes at the option of the user. He may use the program to check his data deck without performing any calculations, or he may execute the program to obtain NEF values. These two modes are called 'noproc' and 'proces' throughout this manual. The names of the mode being identical with the 6-letter keyword which appears in the control card setting the mode. If no specific mode is given, the program assumes 'noproc' and screens data for errors only.

The program goes through three phases during a run. The first phase is the initialization. During this phase the program will perform certain initialization functions such as generating a data base of noise and performance data, etc. The user may build his data base during this phase and set certain options. Since no calculations are performed, only sequence-independent data cards and control cards not dealing with grid manipulations are allowed.

During the second phase, the program will process data cards in either 'noproc' or 'proces' mode until it encounters an END card. The END card signals the end of the data input and calls certain "cleanup" routines before stopping the program execution. During this phase, files which are open are closed, end of information records written where appropriate, etc.

The program calculates NEF values at selected ground observer locations. These ground observer positions are located on a square grid with a 1000-foot spacing between nearest neighbors. These NEF values can be obtained from the computer in printed form, in binary form, or compatible with the California Computer Products, Inc. (CALCOMP) GPCP contouring program. Section VII of this manual is devoted to the interface between NEFUSAF and GPCP, but for a full discussion of the capabilities of the contouring program, one should consult the appropriate CALCOMP manuals. The binary output is only of use to the NEFUSAF program itself when one wants to preserve a particular grid configuration for later use during the run. The printed listing is in the form of a map to the scale of 1:24000 (1" = 2000') and may be printed with or without suppression of empty grid points.

In what is to follow, we will present a more detailed discussion of the output of NEFUSAF followed by a discussion of the three types of input data. In the discussion of input cards, we will present in the text each card with a typical card image followed by its corresponding line printer message. A final section deals with restarting a run which has for one reason or another failed.

SECTION II

OUTPUT, INPUT AND MODE

1. DATA ORGANIZATION

Data can be entered in the program in two ways: cards and binary tapes. Information can be output as binary tape, printed grids or GPCP-compatible files.

The binary tape storage is semi-permanent, in the sense that it can be kept for later use even if in a different run. It is of limited usefulness, however, and it is primarily intended as intermediate storage of results. A binary dump of a grid is also provided when an error is encountered by the program. In that case the binary file can be used to restart the program at a later time when the error has been corrected.

The printed grid is a listing of NEF values at all grid points. It may be kept for reference with plotted grids or it may be used to hand-plot NEF contours. The "normal" output of the program would be the GPCP-compatible output. This output consists of a formatted tape (unit 11) and a set of punched cards (unit 8).

The input prepared by the user is on cards. These are in the form of data cards and control cards. All cards have a 6 character alphabetic keyword. Continuation cards have a blank keyword and a non-blank character must be present in column 79 or 80 of the card preceding it. Only the combination of such a continuation symbol and a blank keyword is recognized as a valid continuation.

Input cards can be separated in 3 types:

1. Sequence independent control cards
2. Sequence independent data cards
3. Sequence dependent cards

This section is concerned with the most basic control functions: how to obtain the printed output and set the program mode. A general discussion of card sequences is also given. For didactic purposes it is advantageous to discuss the output before the input and then present the remaining control functions. The reason is that the engineer ususally has a clear idea of what kind of output is desirable. Once the output options and their use are explained the input functions follow quite naturally. Starting with the input, although at first sight the logical approach, tends to be confusing rather than helpful.

Control cards associated with data input and some further functions are described in Section III. Sequence independent data are discussed in Section IV, while sequence dependent data cards are discussed in Section V where they are presented in an example of an air base. The sequence dependence is developed as a corollary of the normal description of air base operations. The section "Input" of this section gives a formal definition of the allowable card sequences.

Data cards communicating noise and performance data as well as navigational data are acceptable at any time and are sequence independent in the sense that their presence does not affect the sequence prescribed for sequence dependent cards. The user should be aware that sequence dependency in this manual refers solely to whether a card of one kind may follow a card of another type. It does not refer to the requirement that all necessary data must be entered before it is called upon during calculations.

2. FILES USED BY NEFUSAF

A word should be said about file assignments. The program reads cards from unit 5. Unit 6 is assumed to be a printer and contains a listing of all cards encountered as well as other messages. This file is called the "Chronicle" and is discussed below.

Unit 8 is assumed to be a card punch and is reserved for the GPCP interface. Tape unit 11 contains GPCP data and is part of the GPCP interface. During a GPCP run the GPCP program will assume that the data written by NEFUSAF on tape unit 11 is again on tape 11. This enables one to combine a NEFUSAF and a GPCP run in the same job when the standard (positive) options are specified (see Section VII). The nonstandard plotting options (more than 3 cards punched) cannot be run in the same job!

Files available to the user for additional printouts or binary storage are in the range 12-99. Not more than 10 units can be used during a run. The program will not permit binary and formatted dumps to take place on the same logical unit number. Since it serves no purpose and makes tapes unreadable within the scope of the FORTRAN language, one should not attempt to circumvent this protection feature.

Although as is explained later, binary files should never be equivalenced to each other, all printed dumps may be written on the same printer. Since the program prints a separator identifying each printout by unit number and dump number, one can always find the correct dump back. This printer should be logically different from the printer used for the Chronicle to preserve the integrity of the Chronicle.

The logical units which must be assigned as a minimum during the running of NEFUSAF are 3, 4, 5, 6.* If a PLOT card appears

*For certain types of run, unit 4 may not be required. See Section III.3.d.

anywhere in the deck, units 8 and 11 should also be assigned. If a PROCES card appears, unit 10 should be assigned and all units referred to on NEFUSAF I/O control cards. Table II.1 shows the function of all external files.

TABLE II.1

EXTERNAL FILES REFERENCED BY NEFUSAF

Unit	Purpose	Device	Mode	When Needed
3	Scratch Space	Disk	Binary	Always
4	Data Base	Disk	Binary	Always except if NODATA card is present during initialization
5	Control Card Input	Card Reader	BCD	Always
6	Diagnostics ("Chronicle")	Printer	BCD	Always
8	GPCP Control Cards	Card Punch	BCD	If PLOT card is present
10	Default Binary Dump	Tape	Binary	If PROCES card is present
11	GPCP Data Cards	Tape or Punch	BCD	If PLOT card is present
12-99	User assignable	Tape Printer	Binary BCD	If used during 'proces' mode

3. OUTPUT ON UNIT 6: "CHRONICLE"

This unit contains a listing of messages indicating what happened to the data as the data cards were read in. This listing also contains any diagnostics which the program may have occasion to generate; as such the name Chronicle is appropriate.

The Chronicle file is formatted to print 84 columns wide which includes a margin of 10 columns. The reason for this is two-fold: (1) the messages generated do not require the full width of "standard" 14 inch paper; (2) since the Chronicle will, most likely, be kept in the user's file of the particular airbase, a printout the same size as all other documents will be easier to file in the same place.

The Chronicle contains a running account of the processing taking place. There are several general types of entries which are recorded in this file. A first use of the Chronicle is a listing of all control cards as they are encountered in the input. This listing is generally not in card-image format. Rather, appropriate descriptive text is used to augment the data on the cards, making the entry more easily readable and more meaningful in checking the content of the card. An entry in the Chronicle due to reading a card is always preceded by the identifier "+++" in the margin.

A further use of the Chronicle is to provide a place to record certain information generated during the course of the run. There are essentially three levels of information. The first level consists of messages provided "for the record." If the program, for instance, calculates a new altitude versus distance curve during the course of the run, this new curve will be entered in the Chronicle.

The second level of information is provided by the WARNING message. Warnings are signified by a message preceded by a warning banner across the Chronicle page. A summary of warnings is printed at the end of the Chronicle listing for each airfield. A warning is issued at any time when the program detects the existence of a condition where the probability of an error is considerably greater than usual. It is also used to inform the user of the fact that the program has taken a different action than the user had specified because a user command was invalid. The first kind of warning is given, for example, when the user changes the location of a navigational aid; the second type of warning would occur if the user should attempt a binary dump on the line printer.

The third level is the ERROR message. An error occurs when the program detects a condition which will lead to erroneous results or where further processing becomes impossible. Errors are signified by a message preceded by an error banner across the Chronicle page. A summary of errors is printed the end of the Chronicle listing of each airfield. An error would be issued when attempting to process data describing runup operations without specifying where the runup pad is located. When an error occurs, the program switches to the NOGO mode and no further processing takes place for this airfield (see also Section II.7).

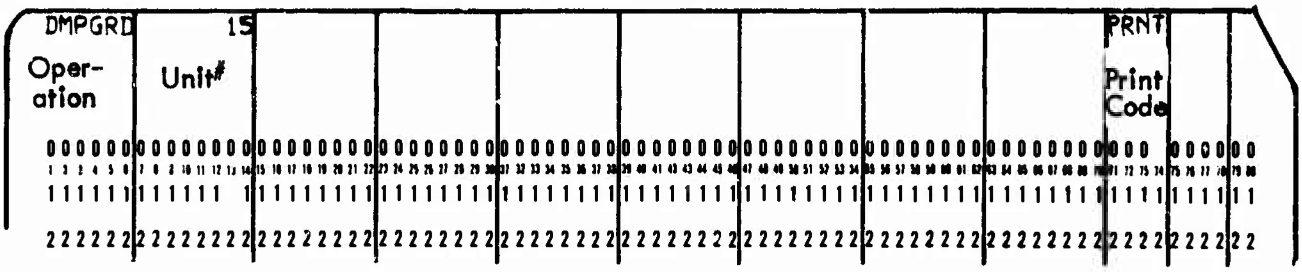
In subsequent sections we will present the NEFUSAF control and data cards and their Chronicle entries. We will present a typical card for each type and follow it with the Chronicle entry which it produces. All diagnostics which each card may generate are also shown as part of the discussion. The Chronicle text for each diagnostic is shown in this manual in the same format as it appears on the printer during a run, except that the values of any alphanumeric field are indicated by the star (*) symbol if they are arbitrary.

4. OUTPUT OF NEF VALUES

After a (cumulative) grid has been computed we must be able to get the data out of the machine. There are three basic ways to communicate the status of the grid to the outside world:

a.	Printed Grid	DMPGRD or PRINT Control Card
b.	Magnetic Tape (binary)	DMPGRD Control Card
c.	GPCP compatible	PLOT Control Card

a. Printed Grid Dump



+++ DUMP GRID TO UNIT 15 PRINTABLE SAVED DUMP *** ON UNIT 15

One way to obtain a picture of the status of the grid is to get a printed listing. For a printed listing the characters PRNT must appear in columns 71-74 of the DMPGRD card. The listing obtained is in a very special format: rather than being merely a listing of the numbers, the "print dump" can be made into a map.

The spacing on the printer is such that the decimal points of all numbers have 1/2-inch separation along the print line (for 0.1" character width), and a 1/2-inch separation is maintained between sucessive lines (for 6 lines per inch printer spacing). This corresponds therefore to a 1" = 2000' scale map of the grid.

The resulting printout allows 5 character. per grid point. Each page will contain 25 points in the X-direction (across the width of the page) and 20 in the Y-direction (down the length of

the page). This fits comfortably on standard 11 X 14 inch computer paper. The printout is then such that a strip from the top to the bottom of the grid is printed on five successive pages; this is followed by the next strip to the right, etc., (Figure II.1)

Page 1	Page 6	Page 11	Page 16
Page 2			
Page 3			

FIGURE II.1 ORDER OF PAGES IN GRID DUMP

If reasonable care is exercised in splicing the printout together to keep the distance of the decimal points of rows and columns to a 1/2-inch between successive sheets, the resulting "digital map" may be used to plot NEF contours by hand. This requires interpolation between neighboring points. In this way, a first estimate can quite easily be obtained. More accurate contouring can be done by hand, but this becomes quite rapidly time consuming. For computer plotting see Section VII, "Noise Exposure Forecast Contours."

The printed grid is useful when there is interest in NEF gradients on the ground or when one is interested in NEF values at locations other than the contours which a computer may have plotted. Another use is in planning studies where the influence of a procedural change on the exposure of a limited area on the ground is to be evaluated. In that case one can compare the printout sheets obtained from several runs which contain the area

of interest. Contours can then be machine plotted for only those cases which seem most promising after a preliminary inspection of the printed values.

NOTE: Grid points for which no NEF value was computed will appear as -88 on the map when a DMPGRD card is used to obtain a printed dump.

The UNIT field specifies the logical unit onto which the grid dump is written. It is the responsibility of the user to see to it that the logical unit selected is assigned to a physical device, and that it is capable of receiving the desired form of output. Since the program is machine independent and written in FORTRAN IV, the program cannot check if a device has been assigned to a logical unit.

Table II.2 shows which UNIT specifications are legal and which fixups are taken if an illegal unit is specified.

TABLE II.2

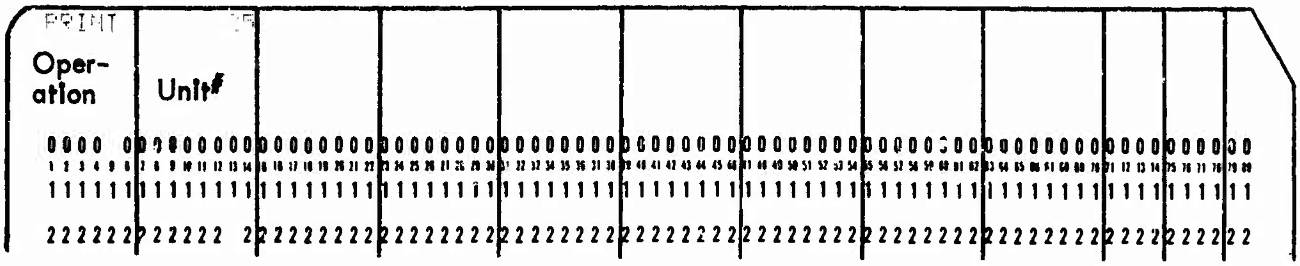
PRINTED GRID UNIT ASSIGNMENT

UNIT Specified	Dump Is Written To Unit	Warning Issued
Blank	6	NO
0	6	NO
1-5	6	YES
6*	6	NO
7-11	6	YES
12-99	as assigned**	NO
>99	6	YES

* Since Unit 6 is a file specially formatted to 84 columns maximum width, a dump there will be printed on 35 pages rather than on 20 pages of 125 columns. These pages are 15 grid values in X wide, except for the last 5 pages which contain only 10 columns.

**For up to 10 logical units referenced, the program will maintain status information. If the unit chosen was used previously for a binary dump, however, a WARNING is issued and the dump is printed on Unit 6.

An alternative way of producing a printed listing of NEF value is by inserting a PRINT card.




```
+++   DUMP GRID TO UNIT 25 PRINTABLE  SAVED DUMP *** ON UNIT 25 SPECIAL FORMAT
```

The PRINT card is used to produce printed grids the same way a DMPGRD card with the PRNT option does, except that empty grid points do not print as -88.0 but are suppressed entirely. Since the format used to print this map must be computed for each individual line of print, this method is more time consuming than the DMPGRD dump. The chronicle message for this type of dump has the extra identification "SPECIAL FORMAT".

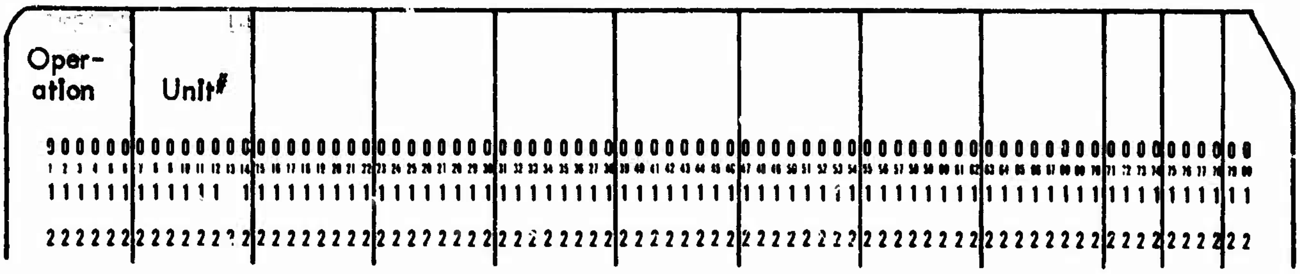
It is recommended that dumps be written on units other than 6 if a map is to be spliced together from a printout, since the 84 character width of the Chronicle causes 35 instead of 20 pages to be printed. If the dump appears in the Chronicle, it will follow immediately after the message:

```
+++   DUMP GRID TO UNIT 6 PRINTABLE
```

or

```
+++   DUMP GRID TO UNIT 6 PRINTABLE  THIS DUMP IS PRINTED IN SPECIAL FORMAT
```

b. Magnetic Tape Dump (binary)



+++ DUMP GRID TO UNIT 14   SAVED DUMP *** ON UNIT 14

If it is desirable to save the current status of the grid for later use by the computer, it can be stored on an external medium such as magnetic tape or any other available serial storage medium. Columns 71-74 of the DMPGPD card should be left blank. The information written on such an external medium is not useful for purposes other than use by NEFUSAF. Again, the logical unit on which one writes a dump must be assigned by the user to the desired device and be capable of receiving binary data. Unit assignment is as follows:

TABLE II.3

BINARY DUMP UNIT ASSIGNMENT

UNIT Specified	Dump is Written to Unit	Warning Issued
Blank	10	NO
1-9	10	YES
10	10	NO
11	10	YES
12-99	as assigned*	NO
>99	10	YES

*For the first 10 logical units referenced, the program will maintain status information. If the unit chosen was used previously for a printed dump, however, a WARNING is issued and the dump is written on logical unit 10.

A binary dump is primarily useful when it is desirable to preserve the current status of a grid for further processing at a later time. This is further discussed in Section III.2. when the control cards for this purpose have been introduced.

c. Messages

If a DMPGRD card references a file which has not yet been referenced, the program will print the message:

PROGRAM CATALOGED FILE AND SAVED DUMP 1 ON UNIT ***

indicating that the file status indicator has been set by the program. In the 'noproc' and 'nogo' mode, the word SAVED is replaced by RESERVED. A maximum of 10 external files may be referenced by NEFUSAF. If more than 10 files are referenced, the message:

FILE CATALOG FULL

is printed. In that case a binary dump will cause an error and a dump to unit 10, and a formatted dump is diverted to the Chronicle after a warning.

Unit 10 serves NEFUSAF as a "safety device." If it is desirable to preserve the status of the grid in binary form and other avenues are not clearly available to the program, the program will write to unit 10. It is recommended that unit 10 always be assigned when the program is in the 'proces' mode since the program will on occasion initiate a dump to unit 10 "on its own." The user may write on unit 10, but cannot read this information back during the same run in which it was written. This is done as a protection since the dump number on unit 10 is not a priori known.

A dump initiated by a DMPGRD card with the unit field unspecified will cause the dump to be written on unit 10. Dumps on unit 10 cannot be read back during the same run that they were written. If the user has no need for the information dumped during the remainder of the job, the use of a blank DMPGRD card will avoid using more tapes than strictly necessary.

From our previous discussion it is clear that there is a fundamental difference between a binary and printed dump. In fact the program will not allow the user to mix both types of output on one logical unit. The program will keep a record of the first 10 legal logical units accessed. This record contains formatted/ binary status information, the current position of the tape (for binary) and the total number of dumps on this logical unit. If more than 10 units are referenced, this status record cannot be kept and an error will result.

The two formats of writing data are incompatible. The computer can read a binary tape, but formatted data cannot under any circumstance be read. On some computer systems it is possible to equivalence two logical units. This should never be done for any binary files referenced by NEFUSAF. When a dump is written the program maintains a count of the dumps to this unit. It is possible to access any one of these during a run provided it was written on a legal unit (12-99). If, for instance, 6 dumps were written onto tape 14, one may then proceed to read dump 4 and after some further processing write a new dump on unit 14. What happens in such a sequence of events? The program will rewind the tape, space over the first 3 dumps and read the fourth one. When a new dump is to be written, the program will space over the next two dumps before writing. If unit 14 was equivalenced with another unit, the dump counts in NEFUSAF would be erroneous. The wrong file could have been read and information beyond the 6th

physical dump would be overwritten. In any event the termination procedure will endfile and rewind each logical unit used for output. The already endfiled and rewound file would then be endfiled again at its loadpoint and all information lost.

NEFUSAF writes an encoded header on each dump taken. An end-of-information record is also present at the end of each binary file which was terminated by NEFUSAF. An attempt to mix binary and formatted data will therefore make the tape unreadable to NEFUSAF. Since NEFUSAF is written entirely in FORTRAN IV, the resulting error message may be a system error rather than an NEFUSAF message. The action will be dependent on the FORTRAN implementation. NEFUSAF will, in the absence of any system error, print the message:

ILLEGAL TAPE HEADER

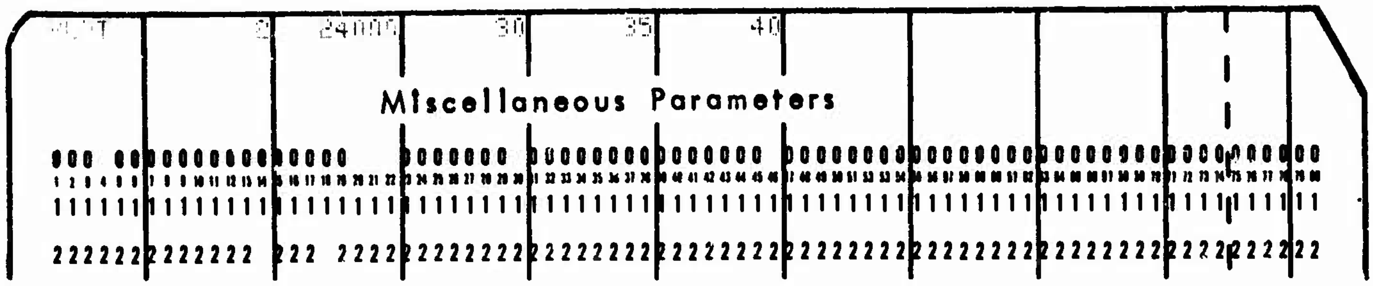
followed by a dump of the information read by the program. The unit will remain inaccessible for the remainder of the airfield considered. When a new airfield is started, the file status will usually* be changed to "input only". Any attempt to get by the bad header will, however, result in the illegal tape header error as before.

This message may also appear after a DMPGRD statement if the unit to be written on was previously used for input only.

---

*If an illegal header appears in a file on which the program has already written, the file remains blocked throughout the remainder of the run.

5. GPCP-COMPATIBLE OUTPUT



Program NEFUSAF computes an array of NEF values. Contours of equal NEF can be obtained by interpolation between the values of this grid. This may be done by hand but is more efficiently done by computer. The program can produce output in such a format that CALCOMP's contouring program GPCP can read the data cards and control cards required to produce contours through GPCP. The process of generating contours with GPCP is fully discussed in Section VII.

The output from NEFUSAF which will later be used by GPCP will mostly appear on logical unit 11 with some output on logical unit 8, which is assumed to be a card punch. The punched cards and the tape from the device assigned to logical unit 11 should be kept together since the combination of these is required for a successful plot. Combining cards of one run with a tape from another run may lead to serious errors.

The file written on unit 11 is formatted to correspond with the normal GPCP input format. Although each record is in card-image format, it is recommended that this file be not punched in cards since each PLOT card may produce up to 11000 records on unit 11.

Furthermore, the deck generated on the punch cannot simply be combined with the tape 11 cards to produce a correct deck of input cards for GPCP. The user wishing to use cards only, is advised to study the CALCOMP manual to find out how the decks should be combined and which cards should be repunched.

6. INPUT

Input acceptable to NEFUSAF consists of sequences of punched cards possibly complemented by one or more binary tapes. Since the cards must somehow convey the complete picture of aircraft operations at an air base, our first task is to develop the particular sequences of cards which will describe an air base.

In the normal course of computing NEF contours there are many different types of information which must be entered. Clearly one needs operational information and runway layout, but in addition many other functions may be required. One can for instance, specify noise and performance data, or the program may be directed to write a dump or create a file of records from which one can plot a set of contours.

The data entered into the computer may conveniently be viewed as statements in a language designed to describe air base operations. This language is very concise, and its syntax rules are very rigorously defined, a feature it has in common with all computer oriented languages. The following section will develop the syntax rules for this language by means of the normal techniques used in language theory.

a. Backus-Naur Form

The concept of a language is useful: it implies that there exist certain fixed syntactic rules which must be followed. Furthermore, these syntactic rules can be quite succinctly and unambiguously stated by the methods commonly used in the description of formal languages. A language used to describe a language is called a metalanguage. In a Latin course for example, English could be used as the metalanguage in which to describe the Latin language. It is not necessary that the metalanguage be different from the language which it describes: the Romans taught their

children Latin using Latin as the metalanguage. A very suitable metalanguage for our discussion here is what is commonly known as "Backus-Naur Form" (or BNF).

A metavariable is designated by enclosing it in square brackets "<" and ">" to distinguish it from a word of the language. (When the metalanguage uses a different alphabet than the language being described, this problem does not occur.) A metavariable is defined by placing it to the left of the metalinguistic symbol "::=", which has the meaning of "is defined as." The portion to the right of the "::=" constitutes the definition of the metavariable being defined. Every metavariable which occurs in the definition of another metavariable, must in turn be defined. This process must therefore be repeated until all metavariables can be described in terms of terminal symbols either by substitution or by deduction.

Each definition as described above is called a "rule" or "production." A production may comprise any number of "alternates," which are separated by "|" (which reads "OR"). Each alternate may have any number of "components" which can be metavariables or letters of the alphabet. The symbol "Λ", which reads "null," is a valid one and denotes an empty component. A production may be recursive, i.e., the metavariable being defined may appear in its own definition. This paradox is resolved by the requirement that any recursive definition shall have at least one alternate which does not contain the metavariable being defined.

With the rules set forth above, one can now unambiguously state which card sequences are syntactically correct in terms of the NEFUSAF language. The productions defined for the language are given in Figure II.2.

One must not confuse syntactic and grammatical correctness with the correctness of the statement itself. It is always possible to write grammatically correct nonsense. The NEFUSAF

```
<NEFUSAF-PKG> ::= <GENERAL-PKG>|<VALID-PKG>|<NEFUSAF-PKG><GENERAL-PKG>|
                  <NEFUSAF-PKG><VALID-PKG>

<VALID-PKG>   ::= <AIRFLD><EXEC-PKG>

<EXEC-PKG>    ::= <EXEC-COMMAND>|<EXEC-PKG><EXEC-COMMAND>

<EXEC-COMMAND>::= <GRID-PKG>|<RUN-PKG>|<FLIGHT-PKG>|<SID-PKG>|
                  <GENERAL-PKG>

<GENERAL-PKG> ::= <INTLZE-PKG>|<GENERAL-COMMANDS>|
                  <GENERAL-PKG><INTLZE-PKG>|
                  <GENERAL-PKG><GENERAL-COMMANDS>

<GENERAL-COMMANDS> ::= <GENERAL-COMMAND>|<GENERAL-COMMAND><GENERAL-COMMANDS>

<GENERAL-COMMAND>  ::= <UNITS>|<PROCES>|<RESET>|<NOPROC>|
                       <WIDTH>|<NODATA>|<END>

<INTLZE-PKG>  ::= <INTLZE-BRANCH>|<INTLZE-PKG><INTLZE-BRANCH>

<INTLZE-BRANCH> ::= <PROFILE-PKG>|<RUN-COMMANDS>|<SIDS>

<PROFILE-PKG> ::= <PROFILE>|<PROFILE-PKG><PROFILE>

<PROFILE>     ::= <ALTUDE>|<DEPNL>|<EPNDB>|<LALTUD>|<LDEPNL>|
                  <LEPNDB>|<XALTUD>|<XDEPNL>|<XEPNDB>

<RUN-PKG>     ::= <RUN-COMMANDS>|<RUN-COMMANDS><RUNUP>|
                  <RUN-PKG><RUN-COMMANDS>|<RUN-PKG><RUNUP>

<RUN-COMMANDS>::= <RUN-COMMAND>|<RUN-COMMANDS><RUN-COMMAND>

<RUN-COMMAND> ::= <RNPPAD>|<RUDSCR>|<PNLT>|<XRUDSC>|<LRUDSC>|
                  <XPNLT>|<LPNLT>|Λ

<FLIGHT-PKG>  ::= <RWY-FTK>|<RWY-FTK><FLIGHTS>|
                  <FLIGHT-PKG><RWY-FTK>|<FLIGHT-PKG><FLIGHTS>

<FLIGHTS>     ::= <FLIGHT>|<FLIGHTS><FLIGHT>

<RWY-FTK>     ::= <RUNWAY>|<RUNWAY><FLTTRK>|<RWY-FTK><RUNWAY>|
                  <RWY-FTK><FLTTRK>|Λ

<GRID-PKG>    ::= <GRID-COMMANDS>|<GRID-PKG><GRID-COMMANDS>

<GRID-COMMANDS> ::= <ADDGRD>|<CLRGRD>|<DMPGRD>|<LODGRD>|
                    <PLOT>|<PRINT>

<SID-PKG>     ::= <SIDS>|<SIDS><DEPART>|<SID-PKG><SIDS>|
                  <SID-PKG><DEPART>

<SIDS>        ::= <SID>|<SIDS><SID>

<SID>         ::= <NAVAID>|<LNAVAI>|<XNAVAI>|<CNAVAI>|Λ
```

Note: In the above each metavariable whose name is identical with an NEFUSAF mnemonic represents the card with keyword. Thus no productions <FLIGHT> ::= FLIGHT appear.

Figure II.2. BNF-Rules for NEFUSAF Cards

routines can check grammatical correctness but only rarely can semantics be analyzed. It is the responsiblity of the user to see to it that the information in the input data correctly reflects the desired air base operation.

b. Organization of the Input

Program NEFUSAF reads the input cards through what amounts to a mnemonic recognizer and syntactic analyzer. The mnemonic recognizer performs a lexicographical scan to see if the OPERATION field of each card is a legal word of the language. A syntactic check is made to see if the appearance of this keyword at this time conforms to the rules of Figure II.2. An out-of-sequence card will result in an ERROR condition. All recognizable mnemonics result in a subroutine call to a routine to check and execute what amounts to the semantics (meaning) of the statement. Since this subroutine call - at least during the PROCES mode - results in the execution of the instruction implied by the card, the language is interpretive.

Although the rules described in Figure II.2 are the complete definition of the syntax rules, there is still considerable work to be done in discussing the semantics of the statements and, in all fairness, a good deal more should be said about the NEFUSAF syntax using English as the metalanguage. The remainder of this chapter is concerned with the Mode of the program. The next chapter deals with further sequence independent control cards; noise and performance data are discussed in Section IV and sequence dependent cards, which describe the aircraft operations, are discussed in Section V.

7. THE MODE OF THE PROGRAM

As was developed in the previous section, the input to the program can be considered as statements in a language designed to describe air base operations. The computer will read these

statements and from this formal description it can compute the Noise Exposure Forecast values associated with the description.

Although it is impossible for the computer to check the semantics of the statements exhaustively (since that would require the computer to know what was required before the data was entered), it is possible to check the syntax of the statements using the computer. In addition, a certain amount of semantic checking can be performed by the machine since it is occasionally possible to detect conflicting or nonsensical meanings in a statement of the language. The NEFUSAF modules have been written in such a fashion that one may check one's entire deck for syntax errors and to some extent for content. It is clear that this process does not eliminate the need to carefully check the data before processing.

On the first pass through NEFUSAF, one can detect all syntactical errors. More than one pass may be necessary to find all detectable semantic errors. If part of a statement is found to contain nonsensical data, this part must be corrected before the rest of the statement can be analyzed because the semantics are highly contextual. In practice this means that the program can detect on the first pass all errors except those which occur in a departure procedure (Section VI). If errors occur in a procedure, it may be necessary to make several passes before all errors are found.

The program provides diagnostics in the form of Warnings and Errors. Errors will stop processing of NEF values, whereas Warnings will allow further processing. Since a warning means that the program has found a condition where the likelihood of a semantic or grammatical error is very high, warnings may often turn out to be errors in preparing the data, therefore: NO DECK SHOULD EVER BE USED TO GENERATE NEF-VALUES UNLESS ALL ERRORS HAVE BEEN CORRECTED AND ALL WARNINGS ARE FULLY UNDERSTOOD.

The program can therefore be run in two different modes: one checks the data decks for errors ('noproc') and one computes NEF values ('proces'). Further a 'nogo' mode is set by the program after an error has occurred. The 'nogo' mode is not selectable by the user. The default mode of the program is 'noproc'. This may at first seem surprising, but is based on the fact that the program will be used more frequently for checking data than for actual processing of air base operations. Furthermore, since the checking can be done quite inexpensively but processing is generally quite time consuming, the user is protected from unintentionally processing data and using large amounts of computer time.

During the 'noproc'ess mode the data cards are checked as explained above. The action implied by the card may or may not be taken, or it may be incomplete. The following rule holds: All operations involving the NEF grid are suspended during 'noproc' mode.

Sequence Independent Control Cards

The grid manipulation cards (ADDGRD, CLRGRD, DMPGRD, PRINT and LODGRD) will print their message but no associated operations will be performed. The program will, however, list most messages as if the action was performed. The Dump messages do not, however, have the key word SAVED, indicating therefore, that the dump was not taken. The CLRGRD card does <u>not</u> have any effect during 'noproc' mode and the last grid will remain unchanged in the machine. The CLRGRD will <u>not</u> result in the message ***GRID CLEARED***. The PLOT card will produce a runway layout only, irrespective of the option selected (See Section VII).

	NOTE: Since dumps referenced during the 'noproc' mode are not read or written, it is not necessary to assign tape drives (or disk space) for such runs. Errors may occur during the 'proces' mode if the data on the tape is found to be inconsistent with the data cards. There is no way to check for this error during 'noproc' mode.
Sequence Independent Data Cards	Sequence independent data cards are processed exactly the same way during 'noproc', 'proces', or 'nogo' mode.
Sequence Dependent Data Cards	These cards will be processed up to the point where NEF values would be calculated. Specifically all data are checked for completeness, and all associated files for plotting are created.

The processing mode is selectable by means of the control card PROCES. Processing of the data will start immediately. That means that if the program was originally in the 'noproc' mode and a PROCES card is encountered, the program will start processing without checking if any of the previous cards should have caused any changes in the values of the NEF Grid. If an error has occurred in the current air field, the 'proces' mode will be deferred until the next AIRFLD card is encountered.

The logical place for a PROCES card is generally only the very first card of the deck, or immediately preceding an AIRFLD card. The only time a PROCES card should be used elsewhere is

during a restart from an error, in which case it is immediately followed by a LODGRD card. All other uses of the PROCES card should be very carefully considered before attempting them.

If decks of 'proces'sable cards and decks of 'noproc'essable cards are combined in a single run, errors witn the file manipulation routines are very likely to occur. It is preferable to have the PROCES decks first and then the data to be checked preceeded by a NOPROC control card. It is further recommended that the binary files referenced by the two data decks be on different logical units.

At the termination of a run the program will print a listing of all files, and how many physical dumps were written to them. This number indicates the real number of dumps written on the volume. The number given as logical dumps indicates the total number of references during 'proces' and 'noproc' modes. A write reference during 'proces' mode to a unit which has been write-referenced during the 'noproc' or 'nogo' mode will, for binary writes, be written on unit 10. The attempt to read this dump will result in an error. It is very difficult to restart this type of error since many cards must be changed in the deck to bring the desired results about. It is therefore recommended to follow the procedure described above and not to mix 'noproc' decks in with 'proces' decks. An example of a termination listing is shown in Figure II.3.

If an error has occurred, the program goes into the 'nogo' mode. The grid values become inaccessible to the user, and to the user the program is essentially in the 'noproc' mode. The PROCES and NOPROC cards will be recognized, but their effect will not take place until a new AIRFLD card is encountered, indicating the start of a new airfield.

Messages resulting from the PROCES and NOPROC cards are:

```
+++  ENTER NON-PROCESSING MODE
     INPUT DATA WILL BE CHECKED BUT NO COMPUTATIONS PERFORMED

+++  ENTER PROCESSING MODE
     CONTOUR COMPUTATIONS WILL BE PERFORMED

+++  ENTER PROCESSING MODE   DEFERRED DUE TO PREVIOUS ERROR
     INPUT DATA WILL BE CHECKED BUT NO COMPUTATIONS PERFORMED

+++  ENTER NON-PROCESSING MODE DEFERRED DUE TO PREVIOUS ERROR
     INPUT DATA WILL BE CHECKED BUT NO COMPUTATIONS PERFORMED
```

```
                  TERMINATION   PROCEDURE                          PAGE     1
----------------------------------------------------------------------------

    FILES REFERENCED DURING THIS RUN

    LOGICAL UNIT 10 CONTAINS    3 DUMPS(PHYSICAL)
    LOGICAL UNIT 20 CONTAINS    0 FORMATTED DUMPS
    LOGICAL UNIT 16 CONTAINS   14 DUMPS(PHYSICAL)   14 (LOGICAL)

    THE GPCP INTERFACE HAS BEEN ENDFILED ON UNIT 11
```

Figure II.3. Typical Termination Procedure Listing

SECTION III

SEQUENCE INDEPENDENT CONTROL CARDS

1. DEFINITION OF SEQUENCE INDEPENDENCE

NEFUSAF can recognize certain control functions which govern details of the operation of the program. These functions deal with the mode of the program as discussed in the previous section, some further options such as the use of metric units, and I/O functions of NEFUSAF. Cards pertaining to such control functions are legal at any time during the execution of the program: they will always be recognized and if they appear in a sequence of sequence dependent cards, the sequence dependency is unaffected by their presence. In other words, if the rules of NEFUSAF specify that a card of type B must follow a card of type A, then the card of type B may follow after any number of sequence independent cards as long as the first one was preceded by a card of type A.*

The cards in this group are distinct from the (also sequence independent) noise and performance data cards, and from those cards which are sequence independent in the sense that they will always be recognized, but which by their presence form the end of one sequence and the beginning of a new one.

Two cards have already been discussed to some extent: PROCES and NOPROC. These cards control the mode of the program. Six cards govern the I/O functions: ADDGRD, CLRGRD, DMPGRD, LODGRD, PLOT and PRINT. The remaining cards govern miscellaneous functions: END, NODATA, RESET, UNITS, WIDTH.

---

*All grid manipulations have meaning only when an airfield is initialized. Therefore an AIRFLD card must have been encountered before any ADDGRD, CLRGRD, DMPGRD, LODGRD, PLOT or PRINT card or an error will result.

2. OPERATIONS ON THE ENTIRE GRID: ADDGRD, CLRGRD, DMPGRD, LODGRD, PRINT

These five control cards govern operations on the grid other than plotting (which is discussed in Section VII).

ADDGRD - add the values of a previously dumped grid to the current grid

CLRGRD - set the grid to zero (-88 NEF)

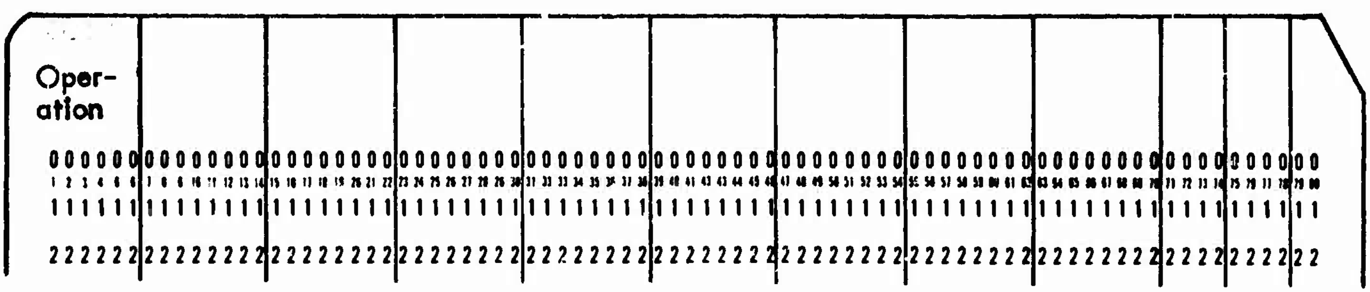
DMPGRD - write a dump to an external file

LODGRD - restore a grid to the value of a previous dump

PRINT - write a formatted dump on an external file and suppress empty points

The simplest operation is the clearing of the grid.

a. The CLRGRD Card




```
+++ CLEAR GRID *** GRID CLEARED ***
```

When the grid is cleared, it is set to a zero value. The NEF sum is logarithmic and since minus infinity cannot be used in a computer, the logical zero of the program is set to -88 NEF. The program in the 'noproc' mode ignores all references to the grid. The CLRGRD card is therefore ignored when the program is in this mode. The value will <u>not</u> be reset until a CLRGRD, or LODGRD card appears <u>after a PROCES card</u>, or until an AIRFLD card is encountered.

A program running in 'proces' mode will upon entering the 'nogo' mode write a dump on unit 10. Dumps on unit 10 are inaccessible to the user during the job. During a later run the same physical volume may be mounted on a different logical unit and be read from that unit. (See Section VIII, Restarting NEFUSAF.) A similar protection feature is invoked when a CLRGRD card is encountered during a 'proces' run. If the then current grid status was not preserved in any form, a dump on unit 10 will be taken. The corresponding message is in both cases:

CURRENT GRID STATUS INACCESSIBLE   SAVED DUMP *** ON UNIT 10

b. The File Catalog

Before proceeding with the cards, which enable the user to take dumps and read them back, we must discuss the way in which the program logic is set up to handle external files. The FORTRAN IV language is rather limited in its I/O capabilities. Although it can write an ENDFILE, no provision is made for recognizing an endfile condition; generally the operating system will abort jobs attempting to read past an end of file. FORTRAN is also not capable of sensing device status and related matters. In order to have at least some control over what external references a user generates,the program will keep information on up to 10 external user-accessible files.

For the first 10 files referenced, the program will maintain a formatting status indicator, a count of the total number of dumps written to the unit, and in the case of binary files, the current position of the read/write head in the file. It is necessary to check that no formatted dumps be written to the same device on which binary information is stored since FORTRAN requires that for binary reads the length of the record read correspond exactly to the record on the storage device. Mixing both kinds of output would therefore make the tape unreadable to FORTRAN programs.

When a reference to an external file is made, the program will, when it first recognizes the file, print the message:

PROGRAM CATALOGED FILE AND

followed by any message which may be appropriate. If more than 10 files in the range 12-99 are referenced, an error

FILE CATALOG FULL

will be issued.

If the program has designated a unit as binary (due to a reference on a DMPGRD, a LODGRD or an ADDGRD card), any subsequent printable dumps attempted on this unit will be diverted to the Chronicle file. If the program has first referenced a file with formatted data, any subsequent binary dumps will be written onto logical unit 10. Any reference on a LODGRD or ADDGRD card to a unit with formatted data will result in an error. The messages can be summarized by:

MODE CONFLICT UNIT IS BINARY .. DUMPED TO UNIT 6

MODE CONFLICT UNIT IS PRINTFILE .. DUMPED TO UNIT 10
SAVED DUMP *** ON UNIT 10

PROGRAM CANNOT READ PRINTFILE BACK

A dump cannot be written to a logical unit with a number of 100 or larger since FORTRAN cannot handle logical unit numbers higher than 99. Also all logical units from 1 through 11 are reserved since they may be used internally by the program or are used by GPCP. Table II.1 lists all NEFUSAF files and their functions.

If an attempt is made by the user to write onto any file other than in the user assignable class, a message appears to indicate the unit chosen by the program:

MANIPULATION WITH ILLEGAL FILE SAVED DUMP *** ON UNIT 10

MANIPULATION WITH ILLEGAL FILE

The last message appears when the dump is written in the Chronicle. Since the program skips to the top of the next page and starts printing the dump no further message is given. The message may also appear if a read reference is made to a dump number less than 1.

We have seen earlier that during 'nogo' or 'noproc' mode, no calculations are performed and no dumps are written. The general philosophy of the program is, however, to give the user all possible information about his data in the 'noproc' mode. This means that dump counts will be printed as if the <u>corresponding dumps had been written</u>. No actual file references take place, however, so that any deck can be <u>checked</u> without the need to have any tapes mounted, except for unit 11 when PLOT cards are present (see Section II.7 and Section VII).

If the user combines decks to be run in 'proces' mode with those run in 'noproc' mode, there exists the decided possibility that the dump counts no longer reflect the actual status of the file. Therefore <u>all files write-accessed during 'noproc' or 'nogo' mode are inaccessible to subsequent DMPGRD control cards which reference them in 'proces' mode</u>. If an attempt is made to

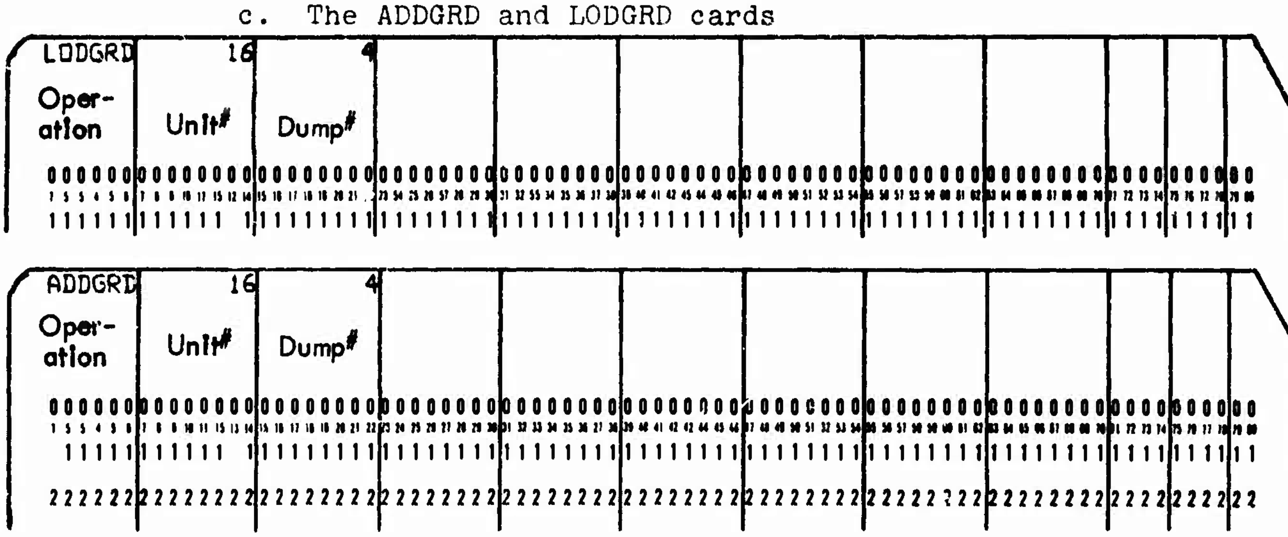
do this, the following message is displayed in the Chronicle:

```
UNIT ** CONTAINS *** LOGICAL BUT ONLY *** PHYSICAL DUMPS
                              SAVED DUMP *** ON UNIT 10
```

The only exception occurs when restarting a job after an error. This is explained in Section VIII.

Further discussion of DMFGRD may be found in Section II.4.

c. The ADDGRD and LODGRD cards



A dump on a user assignable file can be read during the same run in which it was written. The cards shown above direct the reading of the fourth dump on unit 16. When the first read reference is made to a unit on which the program has not written during this run, the warning message:

```
NO KNOWN DUMPS ON THIS UNIT
```

will appear just before the file recognition message. Once the read reference has been established, the program will recognize the tape as an 'INPUT ONLY' file.

The use of binary files allows one to add a previous grid to the current one or to restore the grid to a previous condition. The ADDGRD card will cause the binary dump to be added, the LODGRD card clears the grid before adding the dump. For example, one may have calculated and plotted all flight operations, cleared the grid and calculated and plotted all runups. To make a composite map of runups and flight operations in addition to the individual plots, one can use external files. One can write the "flight" grid on an external file before clearing the grid. After the plotting of the runups, one simply adds the external "flight" grid to the "runup" grid to get the composite.

The user can write more than one dump on any legal unit (see Table II.1) and retrieve it by giving the unit number and dump number. The following few paragraphs outline how one may use this feature, which errors may occur and a few cautionary statements regarding the input/output operations as they relate to the FORTRAN IV language.

If at the time that the request to read is made, the program has written fewer dumps than the number requested, an error will be printed:

```
ONLY *** DUMPS ON UNIT **
```

and the request will be ignored. When a dump number higher than physically present on an 'input only' tape is requested, the program will encounter the NEFUSAF end-of-information record on that unit. The error message will read:

```
ONLY *** DUMPS ON UNIT ** , THE FOLLOWING
DUMP WAS WRITTEN AS END *** ON UNIT ** BY PROGRAM NEFUSAF ON **/**/**
FROM AIRFIELD **** (contents of dump label) ****
```

The program must recognize a valid header on each dump. If the header is not legal, the invalid header is written in the Chronicle:

ILLEGAL TAPE HEADER **** (contents of header) ****

and further access to the file is inhibited until the next AIRFLD card. At that time the status becomes 'input only'. Each subsequent attempt to get beyond the illegal label will cause an error.*

It is possible to write on a tape even if its first reference was a read operation. This is true under any circumstance: a read operation followed by a dump will cause the program to skip over all previously dumped information and write an additional dump. If the initial status of a file was 'input only' and a dump is attempted on this file, the program will locate the end-of-information record and proceed writing. The file status changes from 'input only' to 'input and output' as evidenced by the message:

UNIT ** NOW WRITE ACCESSIBLE.

It should be emphasized that tapes which were written during a run, which terminated due to the operating system action, may not have this end-of-information record and therefore such tapes <u>cannot</u> be read beyond the last dump since ANSI FORTRAN IV cannot

---

*If an unexpected end of information record occurs on a tape on which the program has already <u>written</u>, the unit remains blocked during the rest of the run. This error can only occur if the tape was not positioned at the load point when the job was started, or if logical units were equivalenced to the same physical device.

recognize a tapemark. One should therefore never attempt to add to a tape which was not processed by the NEFUSAF Termination Procedure.* See Section VIII on how such tapes may be salvaged.

When it is desired to extend an already existing file (obtained from a previous run), the logical unit to which it will be associated could be referenced first in an ADDGRD or LODGRD card. If this is not done the first DMPGRD card will cause the previous information to be overwritten. If the file has been referenced in the read mode first, the program will first locate the end record of the last dump written before writing the next dump. When in the contemplated run the first reference is a write, rather than a read, a dummy read reference must be established. This is done by inserting as many ADDGRD cards as are necessary to establish a reference to all tapes which will be augmented. These should preferably be put immediately following the airfield title card, and they must be followed by a CLRGRD to delete the dummy grid which was generated. (If the run which follows starts with a LODGRD to restore a grid which will be updated during this run, the CLRGRD card should of course, be omitted.) Alternatively, one can save time by going into 'noproc' mode, performing the dummy references which now do not transfer information and go back to the 'proces' mode. This way only the "handshake" is performed but no data is transferred, thereby leaving the grid as it was before. The two methods are shown below:

*This rule is somewhat stricter than necessary. Tapes on which the last operation performed during 'proces' mode was read will have a valid end-of-information label and could be augmented exactly as any legal tape.

AIRFLD

	Airfield title card
ADDGRD	(Use as many LODGRD or ADDGRD cards as are required)
CLRGRD	(Optional, see text)
.	
.	
.	Further cards required for this run
.	
.	

Alternatively one can use:

.	
.	
.	Any sequence of cards
.	
.	
NOPROC	
ADDGRD	(As many as required to establish all references)
PROCES	
.	
.	
.	Further cards
.	
.	

In this alternative no information is transferred but only a reference is established between the program and the files.

NOTE: On many computers it is possible to position a tape at the end of the file rather than at the start. Since NEFUSAF does not use the tapemark but a separate FORTRAN-readable record to indicate the end-of-information, this record would then not be erased and remain readable to the program resulting in errors when the files written beyond it cannot be found by the NEFUSAF logic. (Experience has shown that backspacing binary tapes is generally quite unsatisfactory.)

A LODGRD card is entirely equivalent with a CLRGRD followed by an ADDGRD, except when a PROCES card immediately precedes a LODGRD, which is considered as a restart from a previous dump. This is described in Section VIII.

When the program runs in the 'noproc' or 'nogo' mode, the program will print messages indicating the control card directives but no processing is done. During the 'proces' mode the program will print the contents of the header label on the dump to identify the dump being read. Any file positioning done by the program will also be printed. This includes the number of dumps skipped over, if the tape was rewound, and, if the program has the information to calculate the file position. The full message will then be:

```
+++  ADD DUMP *** FROM UNIT **  SKIPPED *** DUMPS AFTER REWIND
     DUMP WAS WRITTEN AS DUMP *** ON UNIT ** BY PROGRAM NEFUSAF ON
     **/**/** FROM AIRFIELD -----------
```

d. The DMPGRD Card

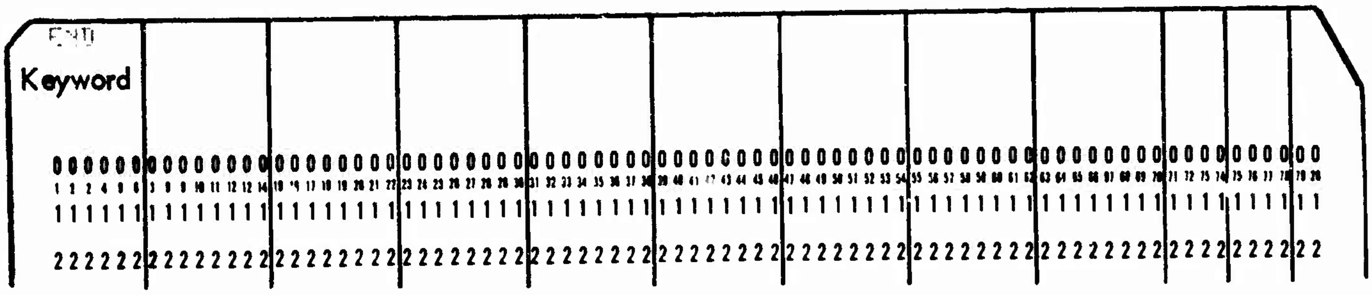
The DMPGRD card has been discussed in Section II.4.

e. The PRINT Card

The PRINT card has been discussed in Section II.4.

3. MISCELLANEOUS CARDS

a. Termination: END



The END card forms the last card in a data deck. It causes the termination of the current airfield much the same as the AIRFLD card does. In addition, the END card causes the proper end-of-information records to be written on all user files. In the case of the plot file for GPCP, all appropriate control cards to terminate a GPCP run will be written on units 8 and 11 if any PLOT cards were encountered during the run.

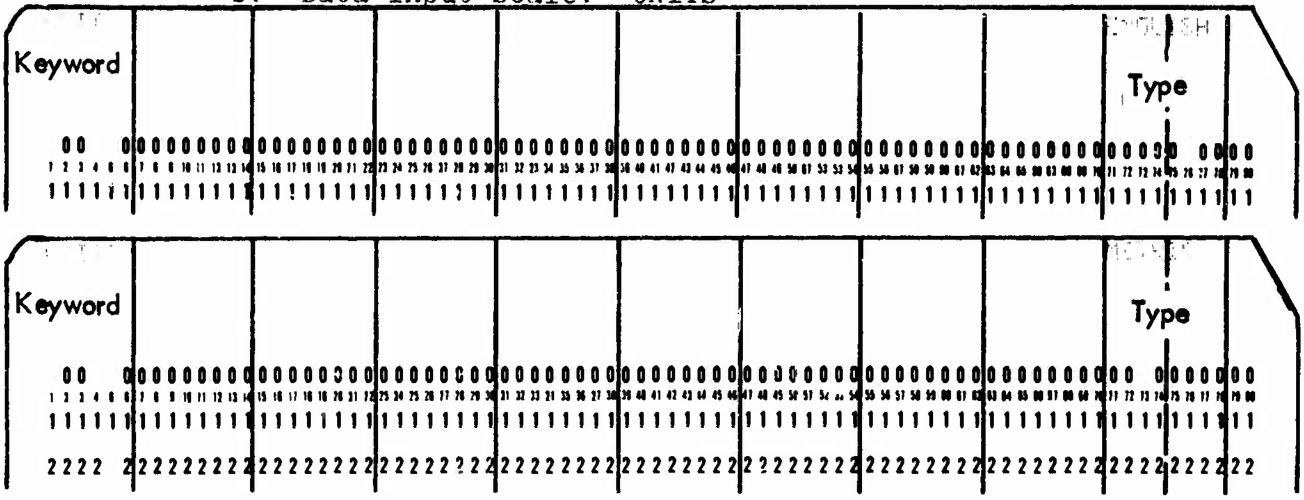
All binary files with 'input only' status are rewound. All files open for "read and write" receive an end-of-information record readable by the program before being rewound. This record is always written if the tape was write accessible and referenced during 'proces' mode. Even if an old end record was present, it will be overwritten with a new one. The entire file status catalog is printed in the Chronicle. If desired, one can tear this section off the main listing and keep it with the reels of tape for later reference.

Only one error can occur due to reading an END card. This happens when the END card is encountered before an AIRFLD card was read. In this unlikely event the message will be:

END CARD ENCOUNTERED DURING INITIALIZATION

As in all cases where an error occurs, the program will terminate abnormally during initialization.

b. Data Input Scale: UNITS



+++ UNITS SPECIFICATION - 'ENGLISH'

+++ UNITS SPECIFICATION - 'METRIC'

The program will by default process all input and output in English units (feet, inches). It is possible, however, to process metric data by using a UNITS card coded METRIC. Although the program internally uses English units, input data will be converted from metric to English units and on output the internal representation will be converted to metric.

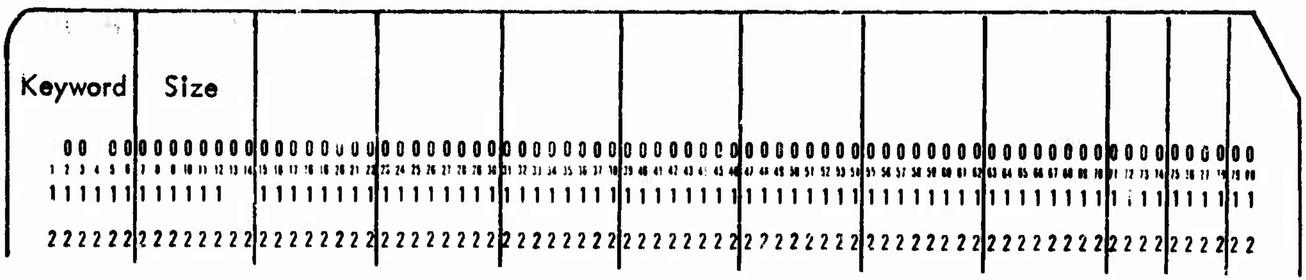
By using UNITS cards with METRIC and ENGLISH at the appropriate places in the deck, it is possible to combine data cards of either type in one deck. One may for instance, use ALTUDE card coded in English units and process a runway layout in metric units, which is convenient when preparing contour maps for areas other than the USA. When the metric option is used, the text "1 inch = ***** FEET" is omitted from the legend on any contour map.

If the specification on a units card is unrecognizable or altogether missing, the program will print the "units" specified followed by the error message:

INVALID UNITS SPECIFICATION - EXECUTION TERMINATED

The units become undefined at this point making further processing meaningless. The card input file is positioned immediately after the END card in the deck. Usual end of job processing by the Termination Procedure takes place.

c. Plotter Paper Size: WIDTH



+++ SET PLOTTER PAPER WIDTH TO 11.0 INCH

If the user does not specify otherwise, the program will assume 28 inches of usable plotter paper width. This reflects the use of a 30 inch drum leaving an inch of border for annotation, etc. The plotter size may, however, be set to a different value.

The size on the WIDTH card is handled differently than all other distances communicated to the program. When the UNITS option is English, the paper width should be in <u>inches</u>; if the metric option is selected the paper width should be in <u>centimeters</u>. In that case the message printed would be:

+++ SET PLOTTER PAPER WIDTH TO ---- CM (**.* INCH)

Irrespective of the units selected, the legend written in the Chronicle for a PLOT card will always give the paper size to the nearest inch.

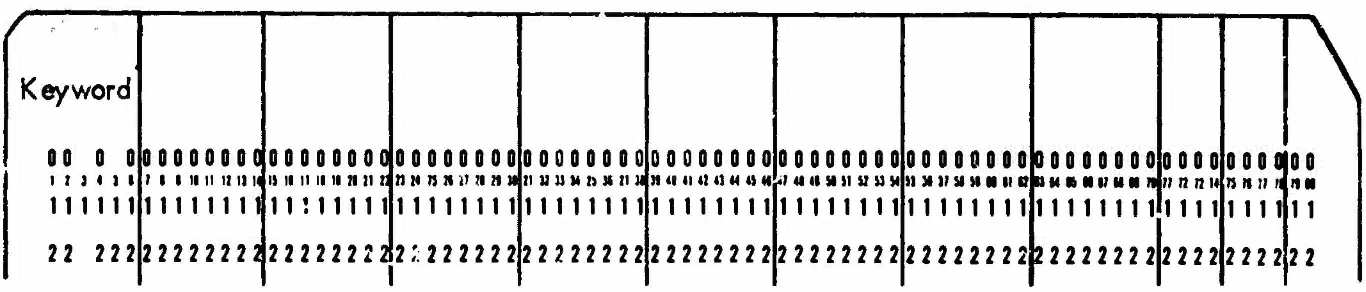
The paper size must be 8 inches or more to be acceptable, otherwise one of the following warnings is issued:

```
PAPER SIZE TOO SMALL, SIZE LEFT AT **.* INCH

PAPER SIZE TOO SMALL, SIZE LEFT AT **** CM (**.* INCH)
```

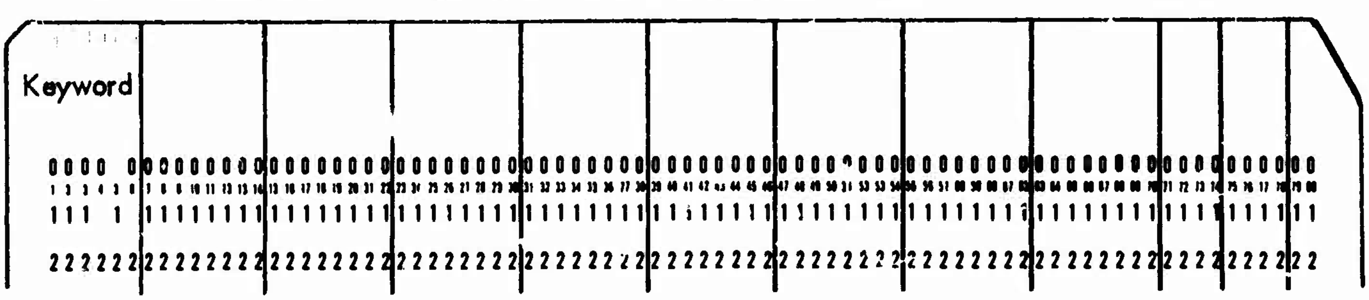
d. Setting and Suppressing an external data base

In order to allow the user to start each airfield with the unmodified data base of the initialization procedure, each AIRFLD card will cause the data base to be reset to its initialization value. This is accomplished by writing the data base on unit 4 when the first AIRFLD card is found and reading this file back at any time a new AIRFLD card is encountered. The user may perform this also at his own discretion by means of the RESET card.




```
+++  RESET DATA BASE TO INITIALIZATION VALUES
```

At times it is undesirable to have the program reset the data base automatically at the start of an airfield. In that case the control card NODATA will suppress this feature.




```
+++  INHIBIT AUTOMATIC RESET OF DATA BASE
```

The inhibit feature is removed as soon as a RESET is encountered. If the NODATA card occurs in the initialization, the program will not write a file on unit 4, thereby limiting the amount of scratch disk space. The word PERMANENTLY is then appended to the usual NODATA message and subsequent RESET cards will cause a NEFUSAF error. A RÉSET card always causes an error if used during the initialization procedure.

SECTION IV

SEQUENCE-INDEPENDENT DATA CARDS

Sequence-independent data cards are those cards which contain noise, performance, or NAVAID information to be used by the program. These cards are sequence independent in the sense of Figure II.2. That is, their presence will not disturb the sequence of sequence-dependent data cards even if interspersed wit . the sequence-dependent cards. The only restriction is that the sequence-independent data must have appeared somewhere ahead of the sequence-dependent cards (FLIGHT and RUNUP) which will use the information contained on the sequence-independent cards.

In order to perform the NEF calculations, the program requires noise and performance characteristics for all aircraft involved in the computations. Rather than supply the program with this information each time an aircraft is "flown" or "runup," pertinent information is numbered and then filed in a data library. Thereafter, the data may be recalled by number when it is required by the program. This section describes the procedures to be used in manipulating the contents of the data library.

The library may be thought of as being divided into two sections:

- data related to aircraft in flight
- data related to aircraft ground runups

Shown below are the seven data sets which comprise the library:

Flight Data	Runup Data
Takeoff Descriptors	Runup Descriptors
Landing Descriptors	PNLT Profiles
Altitude Profiles	
Delta-EPNL Profiles	
EPNL Profiles	

Two data sets are associated with engine ground runups (Runup Descriptors and PNLT Profiles), and five data sets are associated with aircraft in flight (Takeoff Descriptors, Landing Descriptors, Altitude Profiles, Delta-EPNL Profiles, and EPNL Profiles).

The contents of the library may be modified or interrogated by means of appropriate control cards. These control cards allow each of the seven data sets to be manipulated as follows:

- add a new entry
- delete an existing entry
- list all entries

Combining these three manipulating functions with the seven data sets results in a total of twenty-one keyword operators associated with management of the library. These keywords are summarized in Table IV.1.

When running the program, it is necessary only that the data be entered in the library before it is required by a FLIGHT card or a RUNUP card. Also, the order in which the various data sets are updated is completely at the discretion of the user. However, experience has shown that it is desirable to perform all library updating before processing the first airfield. Thus, all data cards dealing with updating the library may be collected in a single deck and placed before the data cards for the first airfield.

TABLE IV.1

LIBRARY MANAGEMENT KEYWORDS

TODSCR - Add the accompanying entry to the TAKEOFF DESCRIPTOR data set.

XTODSC - Delete the accompanying named entry(s) from the TAKEOFF DESCRIPTOR data set.

LTODSC - List all current entries in the TAKEOFF DESCRIPTOR data set.

LNDSCR - Add the accompanying entry to the LANDING DESCRIPTOR data set.

XLNDSC - Delete the accompanying named entry(s) from the LANDING DESCRIPTOR data set.

LLNDSC - List all current entries in the LANDING DESCRIPTOR data set.

ALTUDE - Add the accompanying entry to the ALTITUDE PROFILE data set.

XALTUD - Delete the accompanying named entry(s) from the ALTITUDE PROFILE data set.

LALTUD - List all current entries in the ALTITUDE PROFILE data set.

DEPNL - Add the accompanying entry to the DELTA-EPNL PROFILE data set.

XDEPNL - Delete the accompanying named entry(s) from the DELTA-EPNL PROFILE data set.

LDEPNL - List all current entries in the DELTA-EPNL PROFILE data set.

EPNDB - Add the accompanying entry to the EPNDB PROFILE data set.

XEPNDB - Delete the accompanying named entry(s) from the EPNDB PROFILE data set.

LEPNDB - List all current entries in the EPNDB PROFILE data set.

RUDSCR - Add the accompanying entry to the RUNUP DESCRIPTOR data set.

XRUDSC - Delete the accompanying named entry(s) from the RUNUP DESCRIPTOR data set.

LRUDSC - List all current entries in the RUNUP DESCRIPTOR data set.

PNLT - Add the accompanying entry to the PNLT PROFILE data set.

XPNLT - Delete the accompanying named entry(s) from the PNLT PROFILE data set.

LPNLT - List all current entries in the PNLT PROFILE data set.

This technique has the desirable property that aircraft performance data is physically kept separate from airfield operations data, both in the data card deck as well as in the Chronicle listing.

Obviously special cases will arise where performance data must be modified in the midst of processing an airfield. In such cases, the library management functions may be inserted anywhere in the deck ahead of where the information is needed. Once changes have been made, all subsequent FLIGHT and RUNUP cards will use the updated version of the library (Section III.3.d)

When entering data into the library, it will be noted that distance parameters are called for in certain data sets. These distances may be given either in metric units (meters) or in English units (feet). Unless directed otherwise, the program will assume distances to be specified in feet. If it is desirable to use metric units, this must be communicated to the program via the UNITS specification card. (Section III.3.b) It should be noted that a change in the units specification applies to all input and output for performance data as well as airfield operations data until changed by yet another UNITS card. For the sake of consistency (and to avoid potential errors) it is strongly suggested that the units be consistent throughout the entire performance data package.

During the course of entering data into the library, the program performs a number of checks on the input data. If any errors are found, the program will print a warning message following the listing of the data. Only error-free data is entered into the library, therefore a warning message implies that the entry was not made in the library. Normal processing by the program does continue, however, and only when such data is actually required by a FLIGHT or RUNUP card does the omission become a NEFUSAF error.

As entries are made in a data set, the program will continue to file these entries until all available space has been exhausted. When no further space is available, the program will print the following warning message along with the Chronicle listing of the attempted data entry:

TABLE FULL

Under most circustances, there is ample room to store the performance data which will be required to process a single airfield. If this warning occurs, check the performance data and remove from the card deck those entries which will not be used in performing the calculations for the airfield being processed.

A single card format is used for all library manipulations. In fact, this format is used for all NEFUSAF cards except AIRFLD. The card layout is shown below.

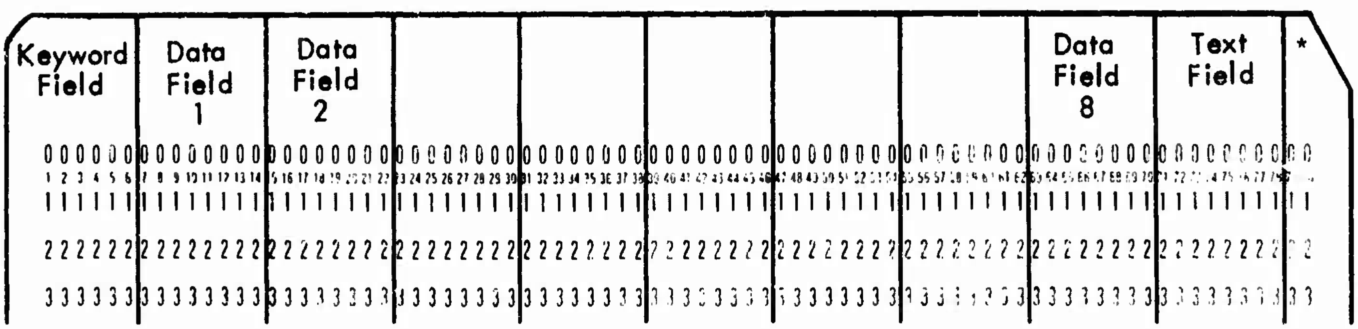


FIGURE IV.1 DATA CARD FORMAT

The first six columns form the keyword field which contains the appropriate keyword (or operation code) for the library function to be performed. The keywords which are less than six characters in length must be left justified in the keyword field, otherwise they will not be recognized by the program.

Columns 7 through 70 are divided into eight, 8-column data fields. These fields are used to enter numeric data. When integers (numbers without decimal points) are entered, they must be right justified in the appropriate field. If numbers contain decimal points, they may be placed anywhere within the field, although right justifying these numbers will improve the appearance of the data card.

Columns 71 - 78 form a text field. This field is used for specifying alphanumeric data such as names of aircraft. Any Hollerith character is legal here.

Columns 79 and 80 form a continuation field. Any nonblank characters in these columns will direct the program to look for additional data cards if there is insufficient room on the first data card to complete the entry. In the remainder of this discussion, we will use an asterisk (*) in column 80 as a continuation character, but this choice is arbitrary.

1. FLIGHT DATA

Before discussing the details of each of the "flight" data sets, it is useful to present a general overview of the functional relationships between the FLIGHT card used during computations and the five "flight" data sets. This is most easily illustrated by tracing through the sequence of data set references once a FLIGHT card has been encountered by the program.

The FLIGHT card contains a two-word numeric name (aircraft number and mission number) to specify the type of aircraft to be flown. The preceding FLTTRK card will have specified whether the flight is a takeoff or a landing. With this information either the takeoff descriptor or the landing descriptor data set is searched for an entry with the corresponding aircraft/mission

numbers. Once the correct entry has been located, the parameter list associated with that entry will contain the numeric names of 1) the altitude profile, 2) the delta-EPNL profile, 3) and the EPNdB profile(s) to be used in the computations. Each of these three data sets are then searched using the aforementioned numeric names to obtain the profile information necessary to perform the computations. Only when the FLIGHT card is encountered is the search performed to verify the presence of each of the needed entries in the various data sets.

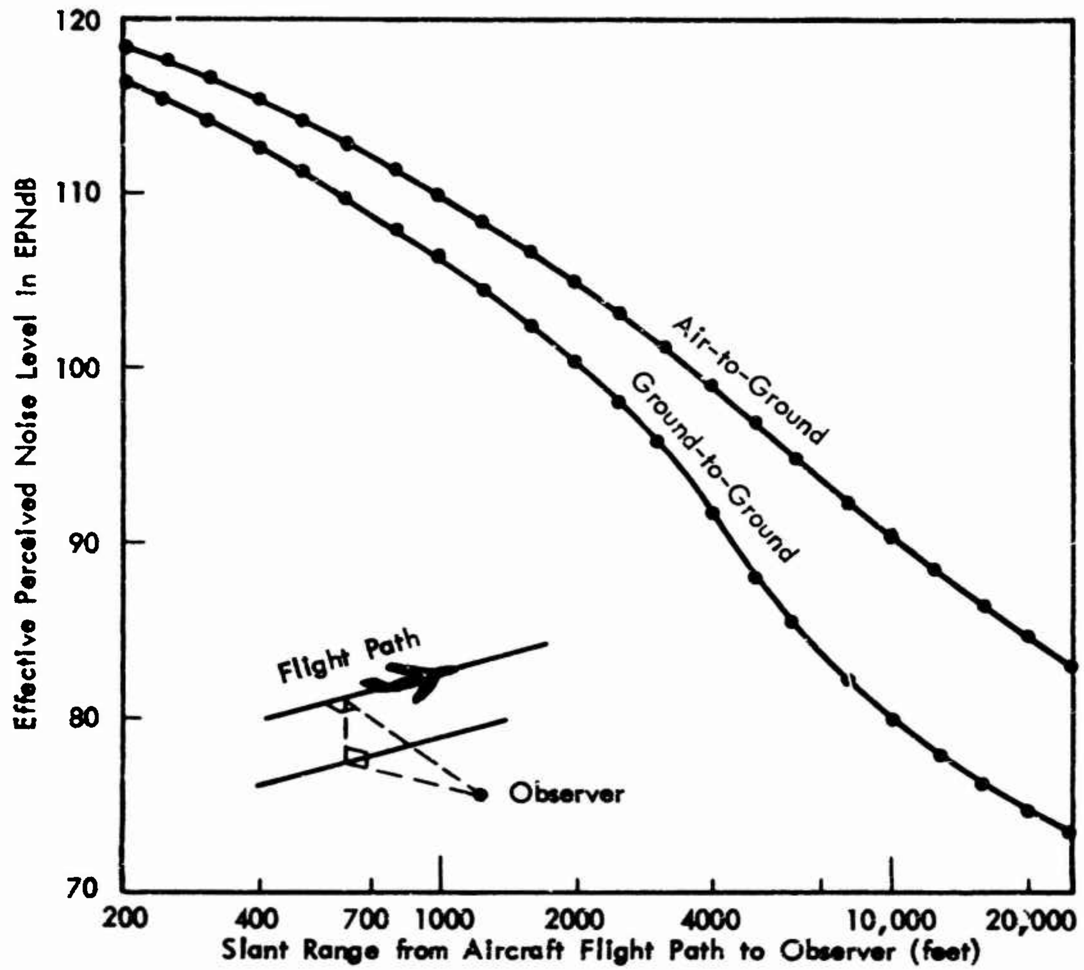
a. EPNdB Profile Data Set

The purpose of the EPNdB profile is to define the aircraft noise exposure level as a function of the slant distance between the observer and the flight path. This profile is constructed by specifying the Effective Perceived Noise Level (EPNL) at a number of fixed distances. These distances encompass a range of 200 feet to 25,000 feet. Figure IV.2 shows a typical profile. Note that the distance is plotted on a logarithmic scale. The 22 fixed distances increase in a fashion such that each distance is 1.259 times as great as the previous one. A complete tabulation of the distances is shown in the figure.

To completely describe the noise propagation characteristics, two profiles are necessary. One profile is used when both the noise source and the observer are on the ground, and is referred to as the "ground-to-ground" profile. The other profile is used when the noise source is in the air and the observer is on the ground, and is referred to as the "air-to-ground" profile. These two profiles together form one complete entry in the data set.

1) Entering An EPNdB Profile

The keyword to be used for entering an EPNdB profile is EPNDB, which is left justified in the keyword field of the data card. Upon recognizing this keyword, the program will print:



EPNL

Distance	G-to-G	A-to-G
200	116.4	118.3
250	115.3	117.8
315	114.0	116.8
400	112.7	115.3
500	111.2	114.1
630	109.7	112.9
800	107.8	111.2
1000	106.2	109.9
1250	104.7	108.3
1600	102.4	106.6
2000	100.3	104.9
2500	98.0	103.0
3150	95.4	101.0
4000	91.8	99.0
5000	88.0	96.9
6300	84.8	94.8
8000	82.0	92.3
10000	80.0	90.1
12500	77.9	88.5
16000	76.0	86.4
20000	74.7	84.5
25000	73.4	82.9

FIGURE IV-2. ILLUSTRATION OF EPNL PROFILE FOR AIRCRAFT IN FLIGHT

+++ FLIGHT NOISE LEVEL PROFILE (EPNL)

This is the first of exactly six data cards which will be required. A typical set of data cards is shown in Figure IV.3.

Data field one contains the numeric name of the profile. The number may be any nonzero, positive integer (no decimal point) and must be right justified in the data field. If a negative number is used, the program will automatically convert it to positive. If zero is used, the following warning message will be printed:

ILLEGAL NAME

The number may be up to 8 digits long, but must be right justified in the data field. The name must be unique among all entries in this data set. If it is not unique, the old profile of the same name will be lost.

Data field two contains a "1" in column 22. This is a propagation code which identifies the next 22 data fields to be the EPNL values making up the "ground-to-ground" profile.

Data fields three through eight contain the first six of these EPNL values. Two additional continuation cards containing eight EPNL values each, are necessary to complete the first profile. Each continuation card must have the keyword field left blank and column 80 of all three cards must have an asterisk (*).

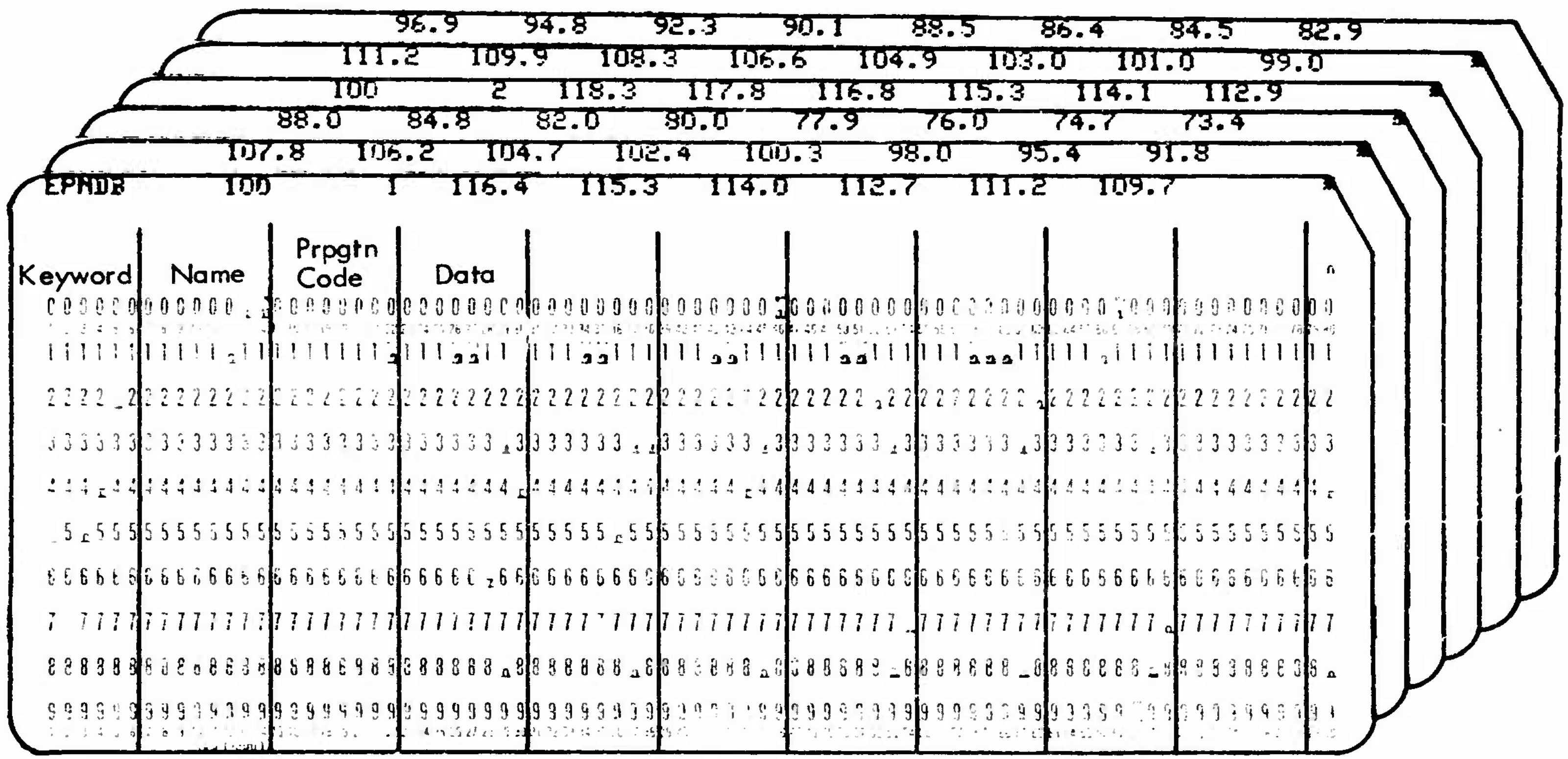


FIGURE IV-3. TYPICAL DATA CARD SET FOR EPNdB PROFILE

The next three continuation cards contain the "air-to-ground" profile. The data format, essentially the same as for the first three cards, is as follows:

The keyword field of the first card is left blank, and data field one must contain the exact same numeric name as appeared with the "ground-to-ground" profile. Data field two must contain a "2" in column 22. This is a propagation code which identifies the next 22 data fields to the EPNL values making up the "air-to-ground" profile. If the numeric names do not agree, or the wrong propagation code appears, the following warning message will be printed.

INVALID NAME AND/OR PROPAGATION CODE NAME = ***** P.C. = *****

Data fields three through eight contain the first six of the EPNL values. The two additional continuation cards contain the remaining sixteen EPNL values. The first two cards in this set must have an asterisk (*) in column 80. The last card does <u>not</u> have an asterisk (*) in column 80.

The program performs a few simple checks for the completeness of the data. Specifically, it checks to see that the first data card is followed by exactly five continuation cards and that the keyword field of each continuation is left blank. If a keyword is encountered on one of the cards the program assumes that one or more data cards were inadvertently omitted, and the following warning message will be printed:

MISSING CONTINUATION CARD

If the program encounters a card without an asterisk(*) in column 80 (other than on the sixth card) the program

will assume that it prematurely encountered the sixth data card and will print the following warning message:

MISSING CONTINUATION CODE OR MISSING DATA

If either of these two conditions should occur, the program will cease to interpret any further cards as belonging to the EPNdB profile. The EPNL values themselves have only two important restrictions. First, the value (which may appear with or without a decimal point) is restricted to plus or minus 200 dB. If the value is outside of this range the comment:

OUT OF RANGE

will be printed next to each offending number and the following warning message will also be printed:

NOISE LEVEL DATA OUT OF RANGE

Second, it is logical that the noise exposure should decrease as the distance between the aircraft and the observer increases. Therefore, consecutive entries in each of the two profiles must be decreasing in value. If this is not the case, the following warning message will be printed and the offending profile will be identified by its propagation code (1 or 2):

NOISE LEVELS NON-DECREASING FOR PRPGTN CODE = **

2) Deleting EPNdB profiles

The keyword to be used is XEPNDB. Upon recognizing this keyword the program will print the following:

+++ EXPUNGE FLIGHT NOISE LEVEL PROFILES

The profiles to be deleted are referenced by their numeric name. Data field one contains the numeric name of the first profile to be deleted, data field two contains the name of the second, and so. Eight numeric names will fit on one card. If more than eight profiles are to be deleted, continuation cards may be added. There is no limit to the number of continuation cards which may be used. However, each continuation card must have the keyword field left blank and an asterisk (*) must appear in column 80 of the preceding card. The last continuation card should not have an asterisk (*) in column 80. The data set is searched for each of the specified numeric names. If the name is found, the profile number is printed. If the name is not found, the profile number is printed along with the phrase:

NOT FOUND

3) Listing the EPNdB Profiles

A complete listing of all entries in the EPNdB profile data set may be obtained by using LEPNDB in the keyword field of the card. Upon recognizing this keyword the program will print the following:

+++ LIST ALL FLIGHT NOISE LEVEL PROFILES

No further data is required on this card.

A complete listing of all profiles in the data set will be printed. Also, if any FLIGHT card refeenced an aircraft whose EPNdB profile was not in the data set, this profile number will be listed with an asterisk (*) printed next to it.

At the end of the listing, the program will print the total number of entries in the data set. This number includes the active entries which were explicitly entered, as well as those which were missing when called upon by a FLIGHT card. In addition, the number of free library entries remaining is printed.

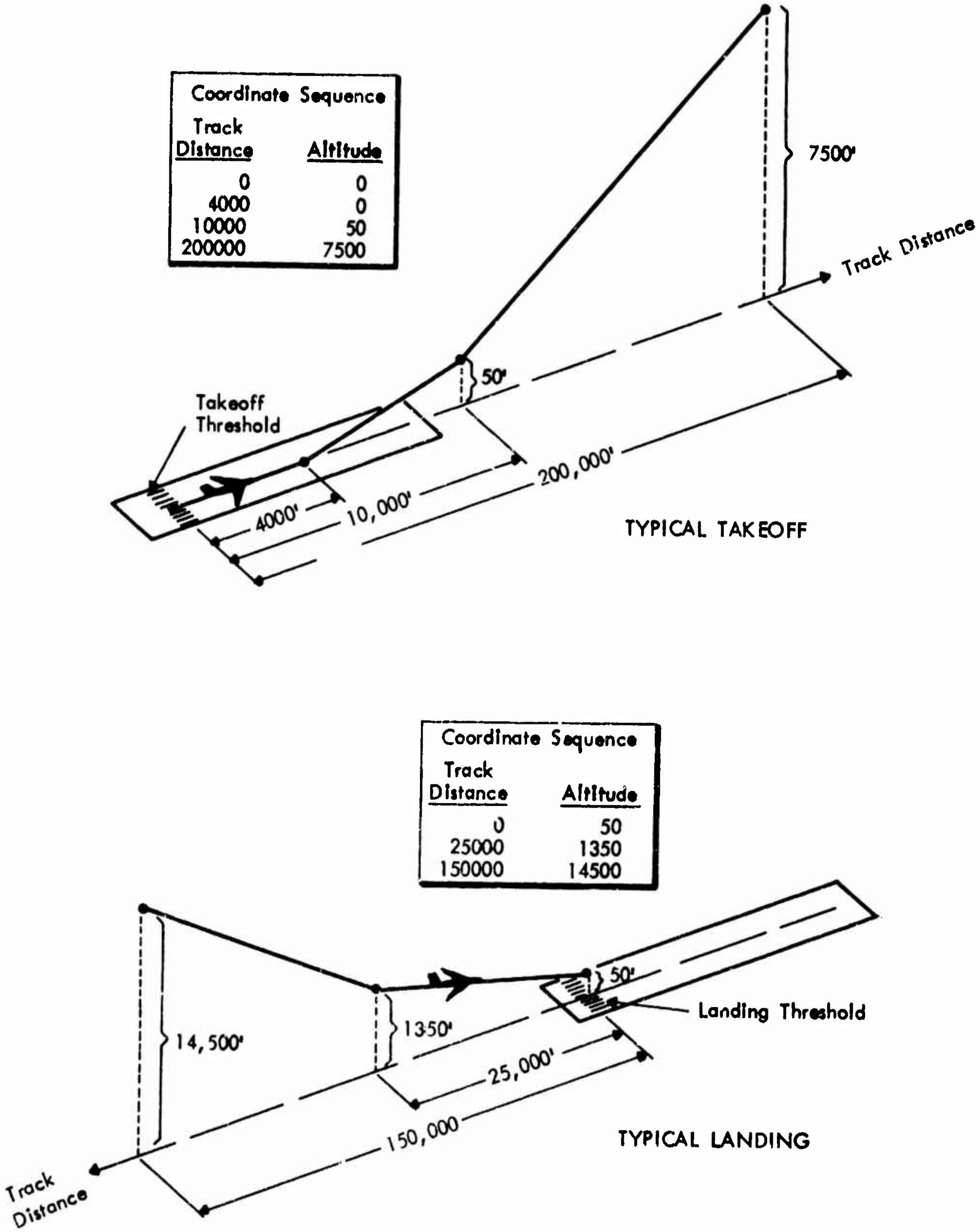
b. Altitude Profile Data Set

The altitude profile depicts the height of the aircraft above ground level versus the flighttrack distance from the airfield. All profiles (whether for takeoffs or landings) are conceived to start at the runway threshold and head away from that runway. The profile is constructed by defining a sequence of (track distance, altitude) coordinate pairs. The program assumes the aircraft to fly along the straight line segments connecting the sequence of points. Figure IV.4 shows a simple altitude profile. The program makes no distinction between profiles used for takeoffs and those used for landings. Only the operational mode (takeoff or landing) of the aircraft which uses the profile determines whether the profile will ultimately be aligned to head down the runway for a takeoff or approaching aircraft. The first coordinate point (which has a track distance of zero) will be aligned with the takeoff threshold for takeoffs and with the landing threshold for landings.

1) Entering an Altitude Profile

The keyword to be used for entering an altitude profile is ALTUDE, which is placed in the keyword field of the data card. Upon recognizing this keyword the program will print the following:

```
+++ ALTITUDE PROFILE
```



Coordinate Sequence

Track Distance	Altitude
0	0
4000	0
10000	50
200000	7500

Coordinate Sequence

Track Distance	Altitude
0	50
25000	1350
150000	14500

FIGURE IV-4. ILLUSTRATION OF ALTITUDE PROFILE

Figure IV.5 shows a typical set of data cards. Data field one contains the numeric name of the profile. This number may be any nonzero, positive integer (no decimal point) and must be right justified in the data field. If a negative number is used, the program will automatically convert it to positive. If zero is used, the following warning message will be printed:

ILLEGAL NAME

The number may be up to 8 digits long but must be right justified in the data field. The name must be unique among all entries in this data set. If it is not unique, the old profile of the same name will be lost.

Data field two is not used. Therefore, leave this field blank. Starting with data field three, the coordinates are entered. Data fields three and four contain the first coordinate pair (track distance and altitude respectively). Succeeding coordinate pairs are entered in data fields five and six, and seven and eight. If more than three coordinates are to be entered, continuation cards may be used. Each continuation card must have the keyword field left blank and an asterisk (*) must appear in column 80 of the preceding card. The last continuation card should not have an asterisk (*) in column 80. Four coordinate pairs may be entered on each continuation card, starting with data field one. The coordinates may be in either feet or in meters. However, be certain that the correct UNITS specification is in effect.

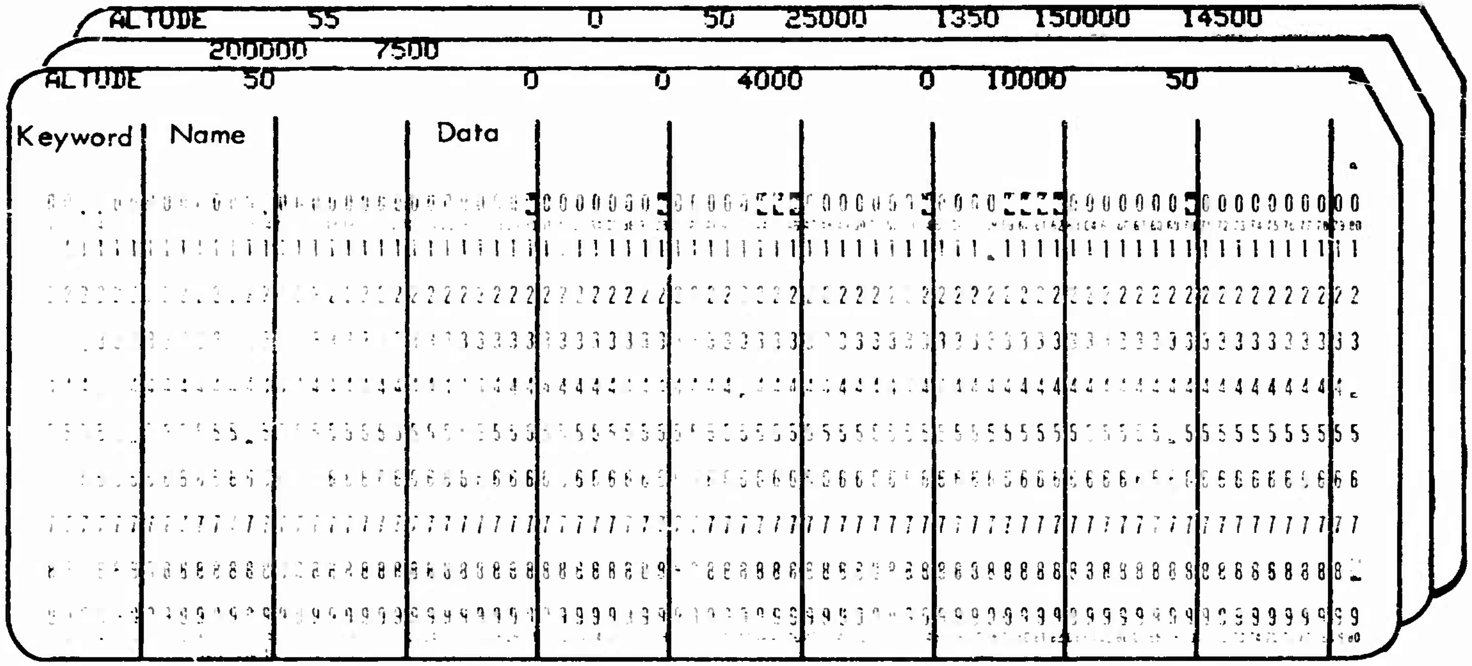


FIGURE IV-5. TYPICAL DATA CARD SET FOR ALTITUDE PROFILE

There are only three restrictions on the coordinates. First, the track distance of the first coordinate must be zero. If it is not, the following warning message will be printed:

```
INITIAL TRACK DIST NOT ZERO
```

Second, all track distances must be positive numbers (with or without decimal points) and must be ascending in value. If they are not, the following warning message will be printed:

```
TRACK DISTANCE(S) NOT POSITIVE OR NOT ASCENDING
```

Third, the number of coordinate pairs must be at least two, but not more than ten. If the number of pairs does not fall in this range, the following warning message will be printed:

```
NUMBER OF COORDINATES RESTRICTED 2 TO 10
```

There are no restrictions on the altitude values. They may be positive, negative or zero.

2) Deleting Altitude Profiles

The keyword to be used is XALTUD. Upon recognizing this keyword the program will print the following:

```
+++  EXPUNGE ALTITUDE PROFILES
```

The profiles to be deleted are referenced by their numeric name. Data field one contains the numeric name of the first profile to be deleted, data field two contains the name of the second, and so on. Eight numeric

names will fit on one card. If more than eight profiles are to be deleted, continuation cards may be added. However, each continuation card must have the keyword field left blank and an asterisk (*) must appear in column 80 of the preceding card. The last continuation card does not have an asterisk (*) in column 80. The data set is searched for each of the specified numeric names. If the name is found, the profile number is printed. If the name is not found, the profile number is printed along with the phrase:

NOT FOUND

3) Listing the Altitude Profiles

A complete listing of all entries in the altitude profile data set may be obtained by using LALTUD in the keyword field of the card. Upon recognizing this keyword the program will print the following:

+++ LIST ALL FLIGHT ALTITUDE PROFILES

No further data is required on this card.

A complete listing of the parameters for all entries in the data set will be printed. In addition, if any FLIGHT card referenced an aircraft whose altitude profile was not in the data set, this profile number will be listed with an asterisk (*) printed next to it.

At the end of the listing, the program will print the total number of entries in the data set. This number includes the active entries which were explicitly entered, as well as those which were missing when called upon by a FLIGHT card. In addition, the number of free library entries remaining is printed.

c. Delta-EPNL Profile Data Set

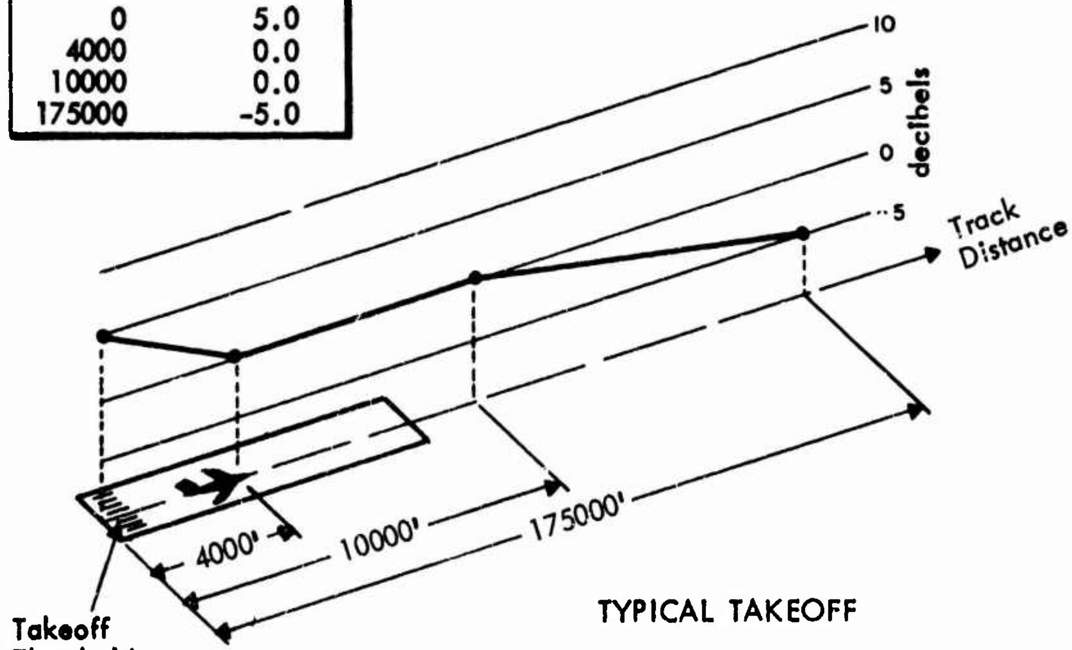
The purpose of the Delta-EPNL profile is to introduce changes in the aircraft noise output as it proceeds through the course of its flight. Frequently an offset, or delta must be added to the noise exposure level specified in the EPNdB profile to account for engine power and speed changes. All profiles (whether for takeoffs or landings) are conceived to start at the runway threshold and to head away from the airfield. The profile is constructed by defining a sequence of (track distance, offset) coordinate pairs. This sequence of coordinates is used by the program to determine the offset level at any point along the flighttrack. Figure IV.6 shows an illustration of a simple Delta-EPNL profile. The program makes no distinction between profiles used for takeoffs and those used for landings. Only the flight descriptor (takeoff or landing) which uses the profile determines whether the profile is used for a takeoff or for a landing, and the two are not mutually exclusive. Although takeoff adjustments are generally numerically different from landing adjustments, the program does not attach a takeoff or landing label to a profile. The first coordinate point (which has a track distance of zero) will be aligned with the takeoff threshold for takeoffs and with the landing threshold for landings.

1) Entering Delta-EPNL

The keyword to be used for entering an altitude profile is DEPNL which is placed in the six column keyword field of the data card. Upon recognizing this keyword the program will print the following:

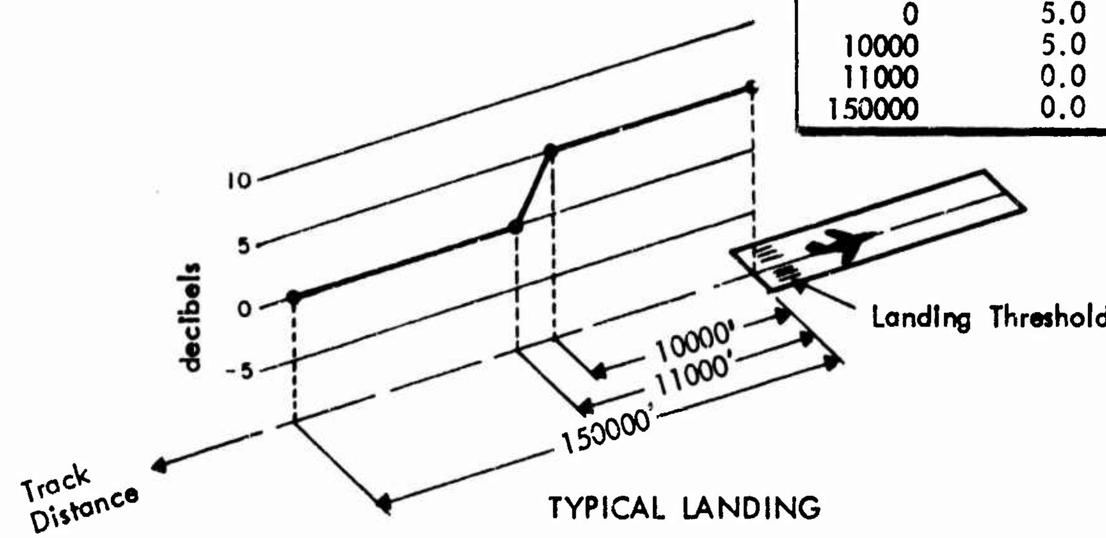
```
+++  POWER LEVEL PROFILE
```

Coordinate Sequence	
Track Distance	Delta
0	5.0
4000	0.0
10000	0.0
175000	-5.0



TYPICAL TAKEOFF

Coordinate Sequence	
Track Distance	Delta
0	5.0
10000	5.0
11000	0.0
150000	0.0



TYPICAL LANDING

FIGURE IV-6. ILLUSTRATION OF DELTA-EPNL PROFILE

Figure IV.7 shows a typical set of data cards. Data field one contains the numeric name of the profile. This number may be any nonzero, positive integer (no decimal point) and must be right justified in the data field. If a negative number is used, the program will automatically convert it to positive. If zero is used, the following warning message will be printed:

ILLEGAL NAME

The number may be up to 8 digits long but must be right justified in the data field. The name must be unique among all entries in this data set. If it is not unique, the old profile of the same name will be lost.

Data field two is not used. Therefore, leave this field blank. Starting with data field three, the coordinates are entered. Data fields three and four contain the first coordinate pair (track distance and offset respectively). Succeeding coordinate pairs are entered in data fields five and six, and seven and eight. If more than three coordinates are to be entered, continuation cards may be used. Each continuation card must have the keyword field left blank and an asterisk (*) must appear in column 80 of the preceding card. The last continuation card should not have an asterisk (*) in column 80. Four coordinate pairs may be entered on each continuation card, starting with data field one.

The distance coordinate may be in either feet or in meters. However, be certain that the correct UNITS specification is in effect. The offset coordinate is always in decibels. There are only four restrictions

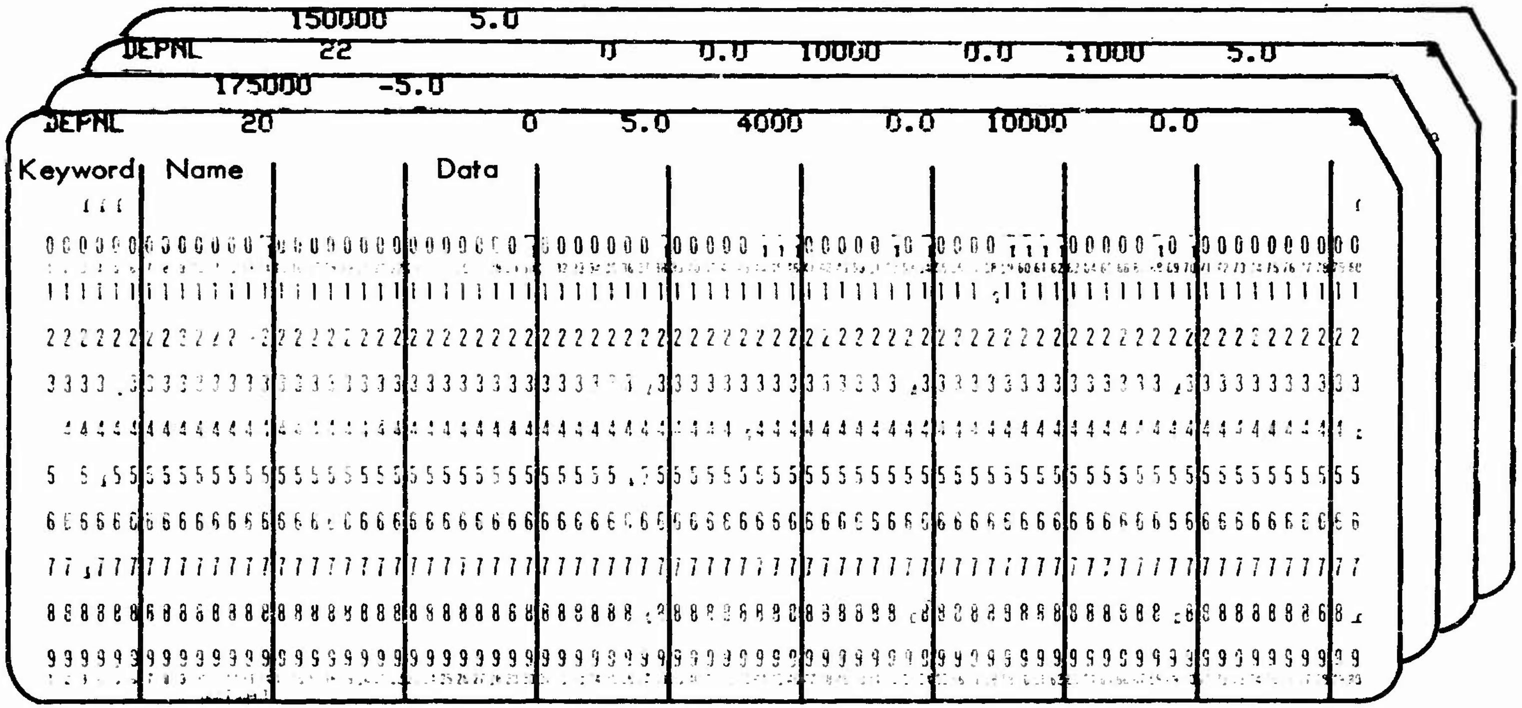


FIGURE IV-7. TYPICAL DATA CARD SET FOR DELTA-EPNL PROFILE

on the coordinates. First, the track distance of the first coordinate must be zero. If it is not, the following warning message will be printed:

INITIAL TRACK DIST NOT ZERO

Second, all track distances must be positive numbers (with or without decimal points) and must be ascending in value. If they are not, the following warning message will be printed:

TRACK DISTANCE(S) NOT POSITIVE OR NOT ASCENDING

Third, the number of coordinate pairs must be at least two, but not more than ten. If the number of pairs does not fall in this range, the following warning message will be printed:

NUMBER OF COORDINATES RESTRICTED 2 TO 10

Fourth, the amount of offset may be positive, negative, or zero . However, the value (which may appear with or without a decimal point) is restricted to plus or minus 200 dB. If the value is outside of this range, the comment:

OUT OF RANGE

will be printed next to each offending coordinate and the following warning message will also be printed:

REL POWER OUT OF RANGE

2) Deleting Delta-EPNL Profiles

The keyword to be used is XDEPNL. Upon recognizing this keyword the program will print the following:

+++ EXPUNGE POWER LEVEL PROFILES

The profiles to be deleted are referenced by their numeric name. Data field one contains the numeric name of the first profile to be deleted, data field two contains the name of the second, and so on. Eight numeric names will fit on one card. If more than eight profiles are to be deleted, continuation cards may be added. There is no limit to the number of continuation cards which may be used. However, each continuation card must have the keyword field left blank and an asterisk (*) must appear in column 80 of the preceding card. The last continuation card should not have an asterisk (*) in column 80. The data set is searched for each of the specified numeric names. If the name is found, the profile number is printed. If the name is not found, the profile number is printed along with the phrase:

NOT FOUND

3) Listing the Delta-EPNL Profiles

A complete listing of all entries in the Delta-EPNL profile data set may be obtained by using LDEPNL in the keyword field of the card. Upon recognizing this keyword the program will print the following:

+++ LIST ALL FLIGHT POWER LEVEL PROFILES

No further data is required on this card.

A complete listing of the parameters for all entries in the data set will be printed. In addition, if any FLIGHT card referenced an aircraft whose Delta-EPNL profile was not in the data set, this profile number will be listed with an asterisk (*) printed next to it.

At the end of the listing, the program will print the total number of entries in the data set. This number includes the active entries which were explicitly entered, as well as those which were missing when called upon by a FLIGHT card. In addition, the number of free library entries remaining is printed.

d. Flight Descriptor (Takeoff and Landing) Data Sets.

The EPNdB, Altitude and Delta-EPNL profiles are tied together by the flight descriptor. As previously mentioned, there are two varieties: Takeoff Descriptors and Landing Descriptors. Their purpose is to specify the noise and performance characteristics of a particular type of aircraft on a particular type of mission. Although the parameter fields on the TODSCR and LNDSCR are layed out exactly alike a logical separation is made between them in order to improve the error diagnostics of the program.

A given mission for a given aircraft will obviously have one specific altitude profile associated with it. Similarly one Delta-EPNL profile is necessary to specify all offsets due to moderate changes in power setting or changes in speed of the aircraft. On the other hand, one may want to specify a completely different EPNdB profile for part of a flight when the power changes are large, such as the use of an afterburner during only part of the mission. Each flight descriptor is therefore allowed to have up to three EPNdB profiles associated with it. Each part of such a mission which has a given EPNdB profile associated with it is called a "subflight." Each descriptor must have at least one subflight specified.

The following parameters are present on a takeoff or a landing descriptor:

- aircraft number } constituting a two-word numeric name
- mission number }
- altitude profile number
- delta-EPNL profile number
- aircraft turn radius
- alphanumeric identifying text
- EPNdB profile number } one pair for each subflight
- applicable track distance }

1) Entering a Takeoff Descriptor or Landing Descriptor

All entries in either data set are made one at a time. The keyword to be used is either TODSCR or LNDSCR depending on whether the flight to be described is a takeoff or a landing, respectively. This keyword is placed in the keyword field of the data card. Upon recognizing either of these keywords the program will print the following:

```
+++  FLIGHT DESCRIPTOR
```

A typical set of data cards is shown in Figure IV.8. The first two data fields of the card contain the aircraft number and mission number respectively (which in combination are referred to as the numeric name). All future references to this flight will be made via this numeric name; therefore, the name chosen must be unique among all entries in the data set in which it is entered. The numbers may be any <u>nonzero</u> positive integer, and must be right

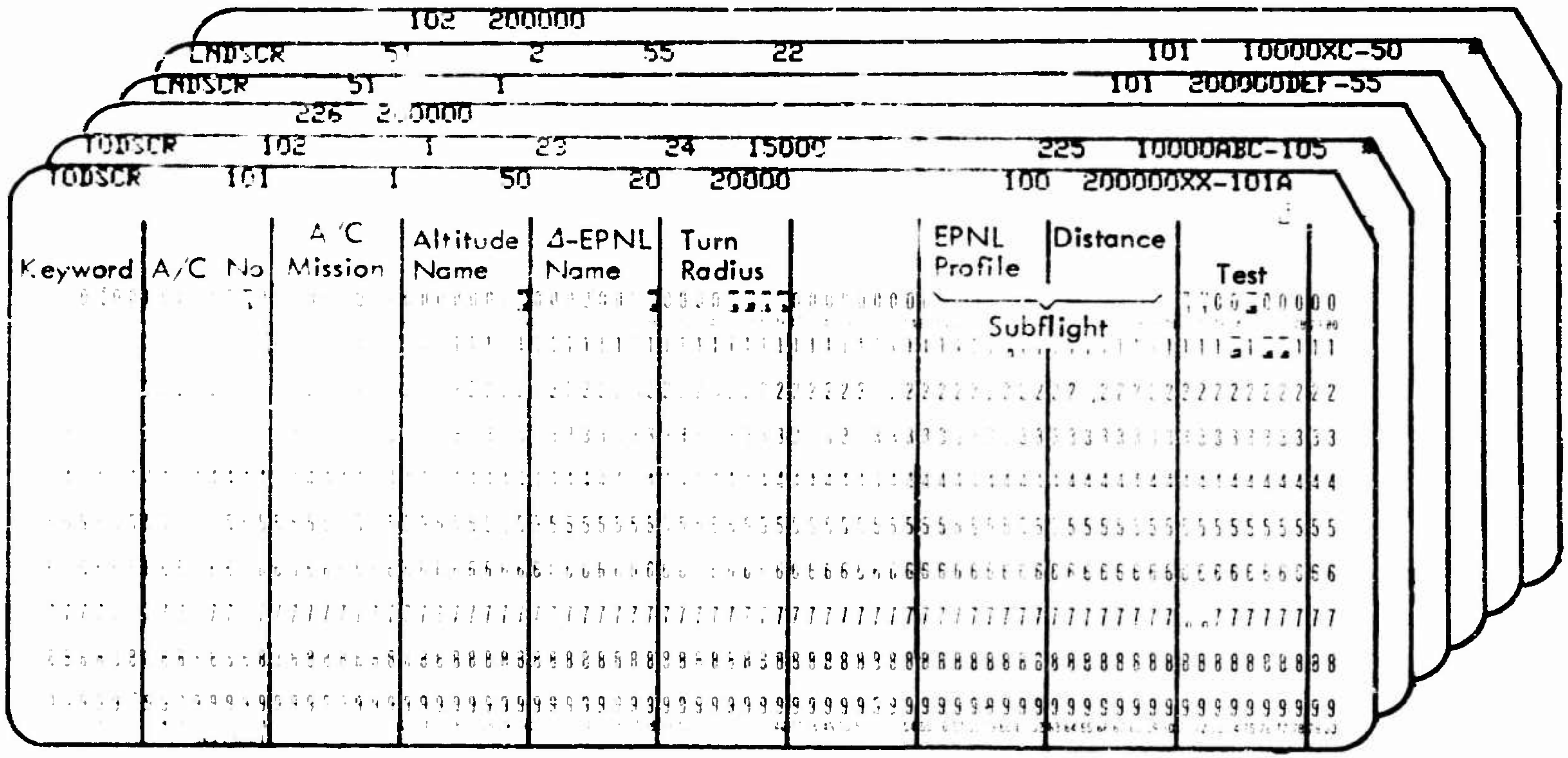


FIGURE IV-8. TYPICAL DATA CARD SET FOR TAKEOFF AND LANDING DESCRIPTOR

justified in their respective fields on the data card. If negative numbers are entered, the program will automatically convert them to positive. If either of the two numbers are zero (or the field is left blank), the entry will not be made into the library and the following warning message will appear:

ILLEGAL AC/MISSION NOS

Data fields three and four contain the Altitude Profile and Delta-EPNL Profile numbers to be associated with this flight. For landings, these fields are normally left blank. By so doing, the program is instructed to compute, 1) its own altitude profile for this aircraft based on a normal glide slope approach to the runway and an assumed 50 foot height over the runway landing threshold and 2) its own Delta-EPNL Profile with zero offset from the EPNdB Profile specified. If the user desires to override this feature he may specify the numeric names of either or both of these profiles (which must then ultimately be entered in their own respective data sets). For takeoffs the program has no means for generating profiles automatically. Therefore, both profile numbers must be specified. By definition they must be nonzero positive integers, and must be right justified in their data fields. Negative numbers will automatically be converted by the program to positive. If zeros are entered (or the fields are left blank), the following warning message will appear:

ALTITUDE OR DELTA-EPNL PROFILE UNDEFINED

The fifth data field contains the turn radius of the aircraft. For takeoff descriptors the turn radius will be

used when the aircraft is flown according to a departure procedure, and is used to generate the flight track when the aircraft is required to make a turn. Thus, this number should be a reasonable estimate of the aircraft turning radius considering the expected performance characteristics of the aircraft within 10 nautical miles of the airfield. The turn radius may be any positive number (with or without a decimal point). If a negative number is entered, the program will automatically convert it to positive. If a zero is entered (or the field is left blank), the program will if the need arises assume a turn radius of 6000 feet (1829 meters). Note that the distance may be specified in either feet or in meters; however be certain that the correct units specification (English or metric) is in effect. When a flight track uses a turn radius smaller than specified on the descriptor card the following message is printed:

```
FLIGHT TRACK TURN RADIUS < ******.* FT AS SPECIFIED ON 'FLTDSC' CARD
```

A flight descriptor must include at least one subflight, but may include up to three. By definition, a subflight is that portion of the total flight over which the character of the noise produced by the aircraft remains constant. That is, the noise propagation from the aircraft can be described by a single EPNL vs. distance function including any offset being introduced by the Delta-EPNL function. Typical situations where a single subflight would suffice are constant power takeoffs and landings. A typical case where two subflights would be desired is an afterburner takeoff. The first subflight would cover that portion of the flight from the start of takeoff roll to the point where the afterburner is secured; the second subflight would start at the point where the afterburner is secured and continue to the end of the flight.

For the first subflight, the EPNdB profile number and distance parameter are placed in data fields seven and eight respectively. The first subflight is assumed to start at a flight track distance of zero* and continues for a distance specified by the distance parameter. The distance parameter is always a positive nonzero number and specifies the total ground track distance traversed by the aircraft to the end of the subflight. The number may appear with or without a decimal point. If a negative number is entered, the program will automatically convert it to positive. The distance may be specified in either feet or meters; however, be certain that 1) the units are consistent with the turn radius in data field five and 2) that the correct units specification is in effect. If only one subflight is sufficient to describe the flight, then be certain to select a high enough value for the distance parameter (say 200,000 feet) so that the program will not cut the flight short in the vicinity of the airfield.

The number of the EPNdB profile to be associated with this subflight is entered in the 7th data field. The actual profile itself is entered into its own data set separately. The number must be a nonzero positive integer, and must be right justified in the data field. If a negative number is entered the program will automatically convert it to positive.

If the flight is to be composed of more than one subflight, the additional data must be entered on a continuation card. This is accomplished by placing an asterisk (*) in column 80 of the first data card to

---

*Track distance zero is the (displaced) landing or takeoff threshold as appropriate.

indicate that a continuation card follows. If the continuation card is inadvertently is omitted, the following warning message will be printed:

MISSING CONTINUATION CARD

On the continuation card, the keyword field is left blank, and data fields one and two are used for the second subflight and data fields three and four are used for the third. By definition, the subflights are sequential; that is, where the first one ends, the second begins, and so on. Thus, the ending track distance for the second subflight must be numerically greater than for the first. Likewise, the third must be greater than the second. If the input data does not conform to this convention, or if a track distance of zero is entered, then the following warning message will be printed:

SUBFLIGHT END DIST MUST BE GREATER THAN BEGIN DIST

The program will not recognize any subflight whose EPNdB profile number is specified as a zero. Furthermore, once a zero EPNdB profile number has been encountered, the program will cease to look for additional subflights. Thus, an EPNdB profile number of zero in the first subflight will result in no subflights being defined. On the other hand, an attempt to enter more than three subflights will overflow the storage capacity of the library. If either of these two conditions should occur, the following warning message will be printed:

NUMBER OF SUBFLIGHTS RESTRICTED 1 TO 3

A text description of the aircraft may also be entered (although it is not mandatory). This description will be printed each time the aircraft is used by the program and provides an easy means for identifying the airplane. The description is entered in the text field of the first data card and may be up to eight characters in length (any Hollerith characters are legal). Although the description need only be entered on the first card, it is recommended that any continuation cards also carry this description in the text field for precautionary purposes.

2) Deleting a Takeoff Descriptor or Landing Descriptor

The keyword to be used is either XTODSC or XLNDSC depending on whether takeoff descriptors or landing descriptors are to be deleted from their respective data sets. Upon recognizing this keyword the program will print the following:

```
++  EXPUNGE FLIGHT DESCRIPTORS
```

Descriptors are referred to by their numeric names (aircraft number and mission number). The first two data fields contain the numeric name of the first descriptor to be deleted, the second two data fields contain the numeric name of the second descriptor to be deleted, the second two data fields contain the numeric name of the second descriptor to be deleted, and so on. Four numeric names will fit on one card. If more than four aircraft are to be deleted, continuation cards may be added. There is no limit to the number of continuation cards which may be used. However, each continuation card must have the keyword field left blank and

an asterisk (*) must appear in column 80 of the preceding card. The last continuation card should not have an asterisk (*) in column 80. For each of the specified numeric names the data set is searched. If the name is found, the aircraft number, mission number, and text descriptor are printed. If the name is not found, the specified aircraft number and mission number are printed, and the phrase:

NOT FOUND

is printed in place of the text descriptor.

3) Listing the Takeoff Descriptors or Landing Descriptors

A complete listing of all entries in the takeoff descriptor and landing descriptor data sets may be obtained by using LTODSC or LLNDSC in the keyword field of the card. No further data is required on this card. Upon recognizing either keyword the program will print the following:

++ LIST ALL FLIGHT DESCRIPTORS

A complete listing of the parameters for all entries in both data sets will be printed. In addition, if any FLIGHT card referenced an aircraft which was not in the data set, the aircraft number and mission number will be listed and an asterisk (*) printed next to the mission number. At the end of the listing, the program will print the total number of entries which were explicitly entered, as well as those which were missing when called upon by a FLIGHT card. In addition, the number of free library entries remaining is printed.

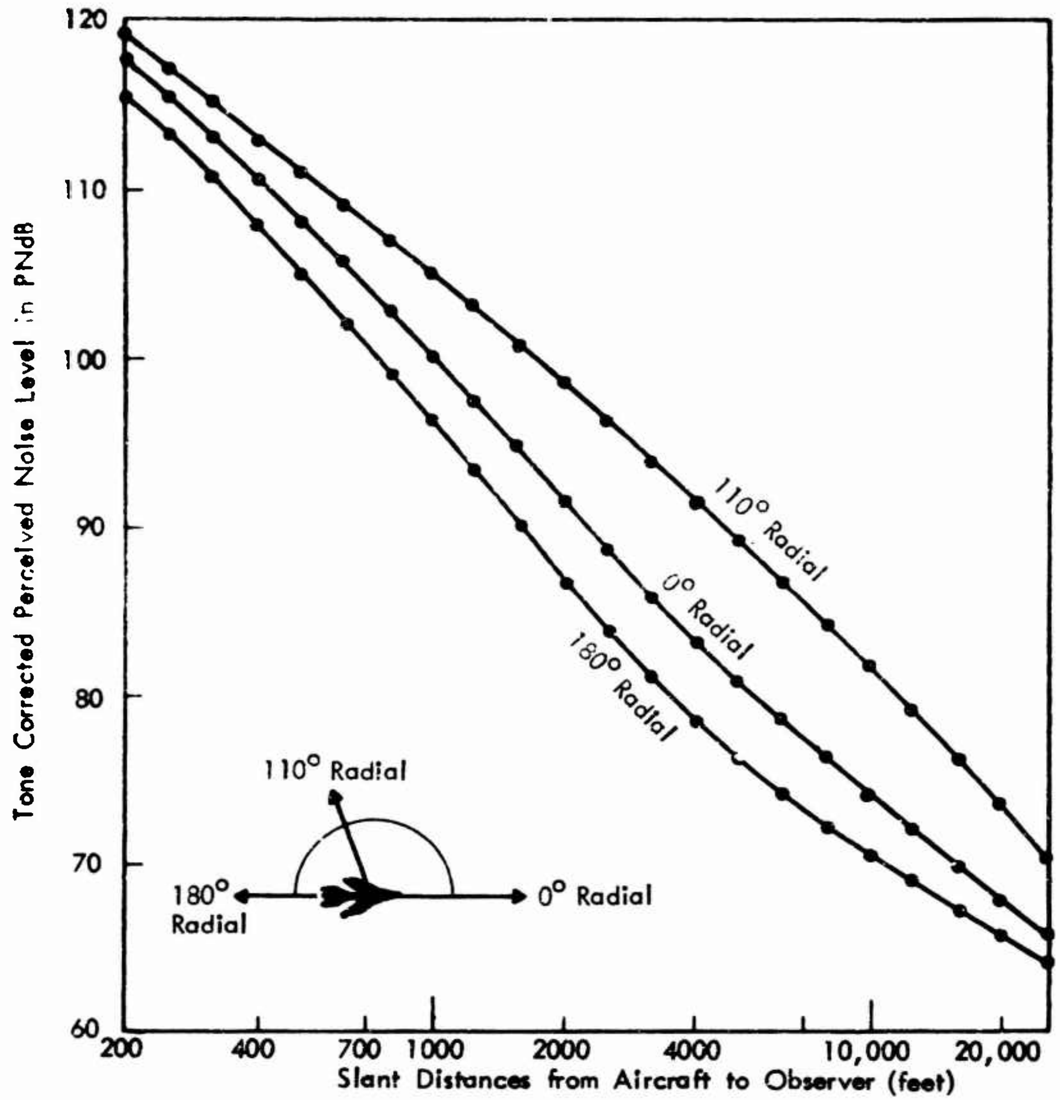
2. GROUND RUNUP DATA

Before discussing the details of each of the "runup" data sets, it is useful to present a general overview of the functional relationships between the runup card used to generate the NEF contribution of a given aircraft and the two "runup" data sets. This is most easily illustrated by tracing through the sequence of data set references once a RUNUP card has been encountered by the program.

The RUNUP card contains a two-word numeric name (aircraft number and thrust) to specify the type of aircraft. With this information the runup descriptor data set is searched for an entry with the corresponding aircraft/thrust numbers. Once the correct entry has been located, the parameter list associated with that entry will make the numeric name of the PNLT profile to be used in the computations available. The PNLT data set is then searched using this numeric name to obtain the profile information necessary to perform the computations. Only when the RUNUP card is encountered is the search performed to verify the presence of each of the needed entries in the various data sets.

a. PNLT Profile Data Set

The purpose of the PNLT profile is to define the noise level of an aircraft during a ground runup operation. The noise level is defined both as a function of the distance from the aircraft and the angle of orientation of the aircraft with respect to the observer. This is done by specifying a number of noise level versus distance profiles at various angles about the aircraft. This is shown graphically in Figure IV.9. Note that the noise levels are only specified for one side of the aircraft. The program assumes that the noise levels are symmetric about the longitudinal axis of the airplane. Therefore, only angles from 0 to 180 degrees are specified.



PNLT

Distance	0°	110°	180°
200	117.5	119.0	115.4
250	115.5	117.2	113.1
315	113.0	115.1	110.8
400	110.5	113.0	107.8
500	108.0	111.0	105.0
630	105.5	109.0	101.8
800	102.7	107.0	98.8
1000	100.0	105.0	96.0
1250	97.4	103.0	93.5
1600	94.1	100.8	89.9
2000	91.5	98.7	86.8
2500	88.5	96.1	84.0
3150	85.7	93.9	81.2
4000	83.0	91.3	78.3
5000	80.6	89.0	76.0
6300	78.5	86.6	74.1
8000	76.1	84.0	72.1
10000	74.0	81.7	70.6
12500	72.0	79.0	69.0
16000	69.7	76.0	67.2
20000	67.9	73.3	65.7
25000	65.8	70.1	64.1

FIGURE IV-9. ILLUSTRATION OF PNLT PROFILE FOR AIRCRAFT GROUND RUNUP

Thus, one entry in the PNLT profile data set is actually a collection of several profiles taken at various angles. Each profile is constructed by specifying the tone corrected perceived noise level (PNLT) at a number of fixed distances. These distances encompass a range of 200 feet to 25,000 feet. The 22 fixed distances increase in a fashion such that each distance is 1.259 (the tenth root of ten) times as great as the previous one. A complete tabulation of the distances is shown in the figure.

1) Entering a PNLT Profile

The keyword to be used for entering a PNLT profile is PNLT, which is left justified in the six column keyword field of the first data card. Upon recognizing this keyword the program will print the following:

```
+++  RUNUP NOISE LEVEL PROFILE (PNLT)
```

A typical set of data cards is shown in Figure IV.10. A set of 3 data cards is required to enter the PNLT profile values for each angle. On the first card, the numeric name of the profile is placed in data field one. This number may be any nonzero, positive integer (no decimal point) and must be right justified in the data field. If a negative number is used, the program will automatically convert it to positive. If zero is used on the first data card, the following warning message will be printed:

```
ILLEGAL NAME
```

The number may be up to 8 digits long but must be unique among all entries in this data set. If it is not unique, the old profile of the same name will be lost.

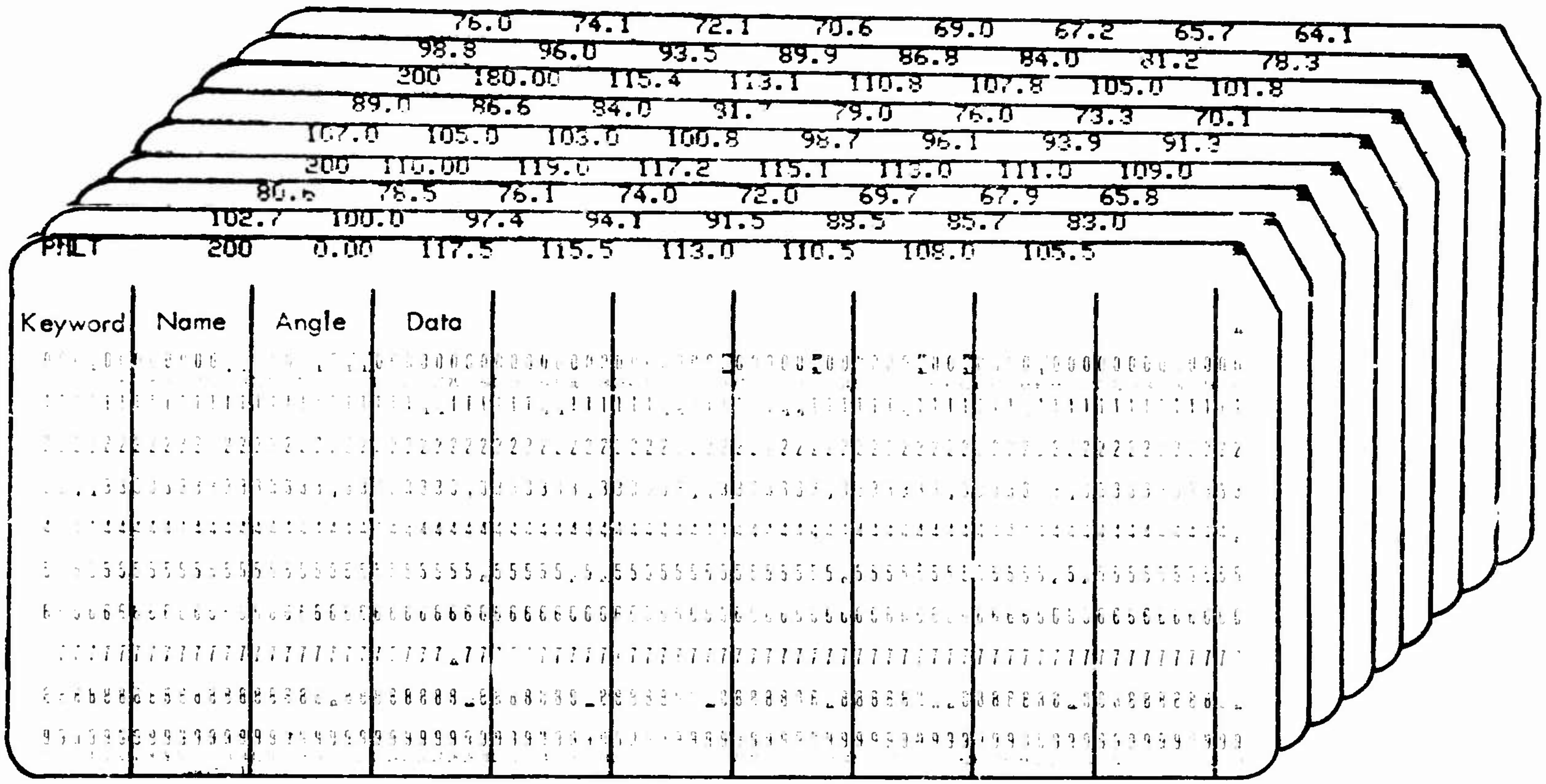


FIGURE IV-10 TYPICAL DATA CARD SET FOR PNLT PROFILE

Data field two contains the angle in degrees. It is mandatory that the first angle entered be zero degrees; thus a zero is placed in column 22 of data field two.

Data fields three through eight contain the first six PNLT values. The additional continuation cards containing eight PNLT values each, are necessary to complete the first profile. Each continuation card must have the keyword field left blank and column 80 of all three cards must have an asterisk (*).

After the profile for the first angle has been entered, the profiles for succeeding angles must be entered in order of increasing angle. A set of three data cards is required for each additional angle and the keyword field on these three cards must be left blank. The format of the three card set is identical with the first angle. The numeric name which is entered in data field one of the first data card in each subsequent set must be identical to the name used in the first set (for zero degrees). If the names do not agree, the following warning message will be printed:

```
NAME DOES NOT MATCH FOR ANGLE = *****
```

In order to aid the user in locating where the problem occurred the message prints the angle which was read from the second data field of this card. If the problem is caused by cards being out of order, the angle value printed may in fact be a PNLT value.

The angle may be any positive number (with or without a decimal point) between 0 and 180. If the number is negative, the program will automatically convert it to positive. Data fields three through eight contain the first six PNLT values. The two additional continuation cards contain the remaining sixteen PNLT values. The PNLT values themselves have only two restrictions. First, the value is limited in magnitude to plus or minus 200 dB. If the value is outside of this range the following warning message will also be printed:

```
NOISE LEVEL(S) OUT OF RANGE
```

Second, it is logical that the noise levels should decrease as the distance between the aircraft and the observer increases. Therefore, consecutive entries in each of the profiles must be decreasing in value. If this is not the case, the following warning message will be printed and the offending profile will be identified by its associated angle:

```
NOISE LEVELS DO NOT DECREASE FOR ANGLE = ***.* DEG
```

The program performs a few simple checks for the completeness of the data. A first check is to verify that 1) there are three data cards for each angle, 2) that all cards have the continuation code (*) in column 80 with the exception of the last card and 3) that the keyword field for each continuation is left blank. If a card with an asterisk (*) is followed by a card with a mnemonic, the program assumes that one or more data cards were inadvertently omitted, and the following warning message, which indicates the last angle read, will be printed:

```
MISSING DATA. LAST ANGLE = ***.*
```

This error can be caused by inadvertently placing an asterisk (*) in column 80 of the third card of the last angle, thus causing the program to expect another profile for that entry.

Once the program encounters a card with column 80 left blank, it assumes that this card terminates the data entry. Clearly this is a critical problem if column 80 is left blank on either of the first two cards of the three card sequence for a particular angle. If such is the case, the program assumes that either the continuation code (*) was omitted from this card or that some data cards are missing. It then prints the following warning message

MISSING CONTINUATION CODE OR MISSING DATA. LAST ANGLE = ***,

If either of these two error conditions should occur, the program will cease to interpret any further cards as belonging to the PNLT profile.

Additional checks involve the angles which have been specified. If the profiles are not in ascending order of angle, or if two profiles are specified with the same angle, the following warning message will be printed:

ANGLES NOT IN ASCENDING ORDER OR DUPLICATE ANGLE

It is possible (but very unlikely) that this message could be manifested by cards being out of order.

Profiles for up to 10 angles may comprise one entry. If more than ten are specified, the listing of the input data will show only the first nine angles and the last

angle (rather than the tenth angle) entered, and the following warning message will be printed:

TOO MANY ANGLES, 10 MAX

2) Deleting PNLT Profiles

The keyword to be used is XPNLT which must be left justified in the keyword field. Upon recognizing this keyword the program will print the following:

+++ EXPUNGE RUNUP NOISE LEVEL PROFILE

The profiles to be deleted are referenced by their numeric name. Data field one contains the numeric name of the first profile to be deleted, data field two contains the name of the second, and so on. Eight numeric names will fit on one card. If more than eight profiles are to be deleted, continuation cards may be added. There is no limit to the number of continuation cards which may be used. However, each continuation card must have the keyword field left blank and an asterisk (*) must appear in column 80 of the preceding card. The last continuation card should not have an asterisk (*) in column 80. The data set is searched for each of the specified numeric names. If the name is found, the profile number is printed. If the name is not found, the profile number is printed along with the phrase:

NOT FOUND

3) Listing the PNLT Profiles

A complete listing of all entries in the PNLT profile data set may be obtained by using LPNLT in the keyword field of the card. Upon recognizing this keyword the program will print the following:

+++ LIST ALL RUNUP NOISE LEVEL PROFILES

A complete listing of the parameters for all entries in the data set will be printed. In addition, if any RUNUP card referenced aircraft whose PNLT profile was not in the data set, the profile number will be listed with an asterisk (*) printed next to it.

At the end of the listing, the program will print the total number of entries in the data set. This number includes the active entries which were explicitly entered, as well as those which were missing when called upon by a FLIGHT card. In addition, the number of free library entries remaining is printed.

b. Runup Descriptor

The purpose of the runup descriptor is to associate a noise level versus distance profile set with a particular aircraft. A runup descriptor is comprised of the following five basic parameters:

.aircraft number } two-word numeric name
.thrust number }

.PNLT profile number

.PNLT offset

.alphanumeric text description

1) Entering a Runup Descriptor

All entries in this data set are made one at a time. The keyword to be used is RUDSCR. Upon recognizing this keyword the program will print the following:

+++ RUNUP DESCRIPTOR

Typical data cards are shown in Figure IV.11. The first two data fields of the card contain the aircraft number and thrust number, respectively, which in combination are referred to as the numeric name. All future references to this runup will be made via this numeric name; therefore, the name chosen must be unique among all entries in the data set. The numbers may be any nonzero positive integer, and must be right justified in their respective fields on the data card. If negative numbers are entered, the program will automatically convert them to positive.

Data field three contains the number of the PNLT profile to be associated with this runup. Only the numeric name of the profile is entered as a part of the runup descriptor. By definition it must be a nonzero positive integer, and must be right justified in the data field. A negative number will automatically be converted to positive.

If zeros are entered in any of the first three data fields (or any are left blank) the entry will not be made into the library and the following warningmessage will be printed:

INVALID AC CLASS, THRUST, OR PNLT PROF

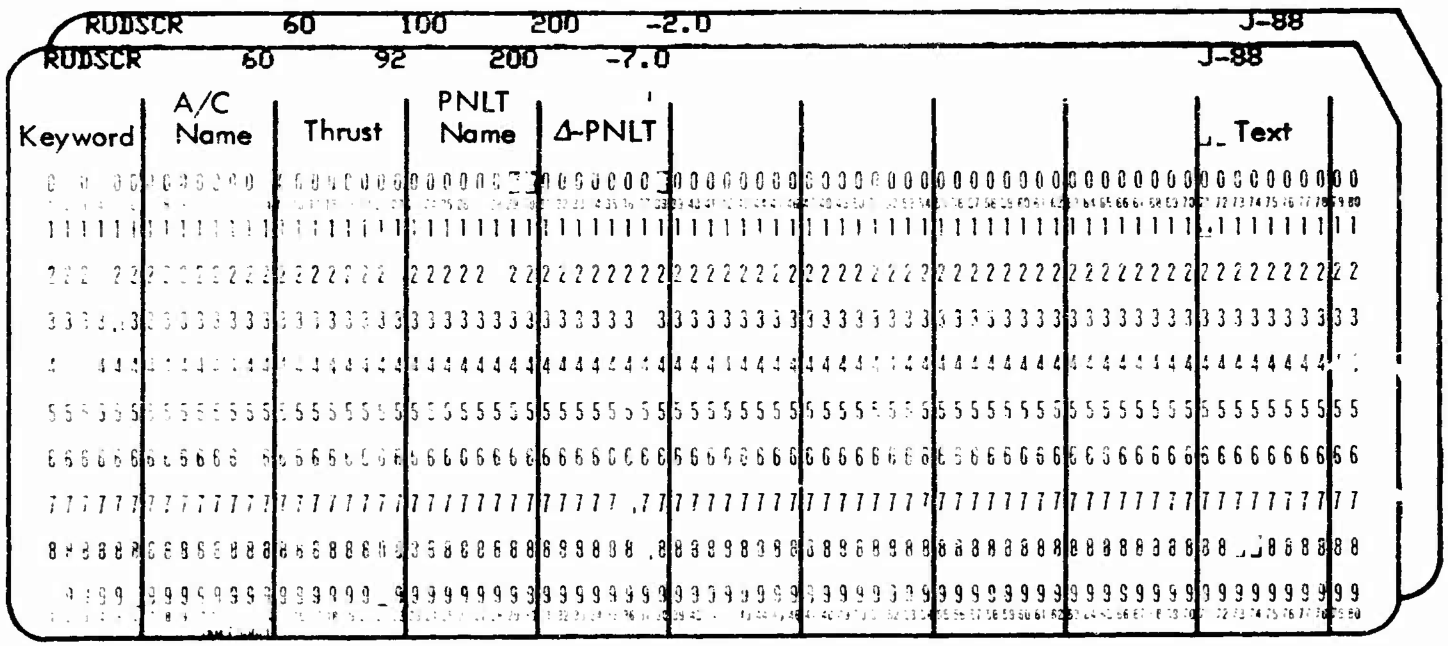


FIGURE IV-11 TYPICAL DATA CARD SET FOR RUNUP DESCRIPTOR

The fourth data field contains the PNLT offset. Frequently, several aircraft will produce similar noise level patterns, the only difference being the magnitude of the level. In such cases, a single PNLT profile may suffice for these aircraft. The offset specifies the value in decibels which will be added to the PNLT values to determine the ultimate noise level of the aircraft. The number (with or without a decimal point) may be positive, negative, or zero, but is limited in range to plus or minus 200 db. If the value entered on the data card is outside of this range, the following warning message will be printed:

OFFSET OUT OF RANGE

A text description of the aircraft may also be entered (although it is not mandatory). This description will be printed each time the aircraft is used by the program and provides an easy means for identifying the airplane. The description is entered in the text field of the data card and may be up to eight characters in length (any Hollerith characters are legal).

2) Deleting a Runup Descriptor

The keyword to be used is XRUDSC. Upon recognizing this keyword the program will print the following:

+++ EXPUNGE RUNUP DESCRIPTORS

Descriptors are referred to by their numeric names (aircraft number and thrust number). The first two data fields contain the numeric name of the first descriptor to be deleted, the second two data fields contain the

numeric name of the second descriptor to be deleted, and so on. Four numeric names will fit on one card. If more than four aircraft are to be deleted, continuation cards may be added. There is no limit to the number of continuation cards which may be used. However, each continuation card must have the keyword field left blank and an asterisk (*) must appear in column 80 of the preceding card. The last continuation card should not have an asterisk (*) in column 80. For each of the specified numeric names the data set is searched. If the name is found, the aircraft number, thrust number, and text description are printed. If the name is not found the specified aircraft number and thrust number are printed, and the phrase:

NOT FOUND

is printed in place of the text description.

3) Listing the Runup Descriptors

A complete listing of all entries in the runup descriptor data set may be obtained by using LRUDSC in the keyword field of the card. Upon recognizing this keyword the program will print the following:

+++ LIST ALL RUNUP DESCRIPTORS

A complete listing of the parameters for all entries in the data set will be printed. In addition, if any runup card referenced an aircraft which was not in the data set, the aircraft number and thrust number will be listed and an asterisk (*) printed next to the thrust number.

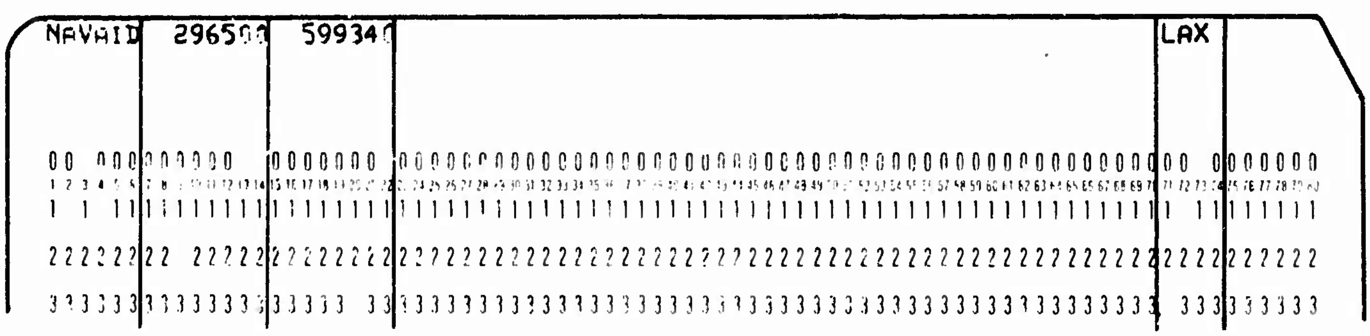
At the end of the listing, the program will print the total number of entries in the data set. This number includes the active entries which were explicitly entered, as well as those which were missing when called upon by a RUNUP card. In addition, the number of free library entries remaining is printed.

3. NAVIGATIONAL AID DATA

When departure procedures are used in the data such procedures may reference navaids. Just as any noise and performance data does not cause an error unless a FLIGHT or RUNUP card calls for the information, navaid information will only cause an error if a FLIGHT card causes the execution of a procedure which calls for the information.

a. Entering a Navaid

The keyword to be used for entering a navaid is NAVAID.



+++ ENTER NAVAID LAX AT X = 296500. Y = 599340 FT

The navaid coordinates referred to the map used for entering airfield geometry (see Section V) are placed in data fields one and two. The navaid identifying code is entered in the first alpha field (columns 71-74). It is important to recognize that a blank space in this field has meaning. If the usual three

letter codes are used, there is a difference between ' LAX' and 'LAX '. To the program these are distinct entries and one should take care to consistently left or right justify the navaid codes on NAVAID cards as well as DEPART cards. Only one navaid can be entered per card, but continuation cards may be used.

If more navaids are entered than the program can store, the program will print the warning:

TABLE FULL

Although it will continue reading and listing cards, they will be ignored. If a new entry is made with the same alphanumeric code as an existing entry, the program will issue a warning:

A PREVIOUS ENTRY FOR **** HAS BEEN DELETED

The reason for printing a warning rather than a simple message as is done in noise and performance data is that, although modifying operational parameters during the calculation for an airbase is fairly routine, moving navaids around is at the very least a suspicious activity.

Any reference to a navaid must include a nonblank entry in columns 71-74. If a blank identification code is used, the program cannot retrieve the information and the entry is lost. This is signified by the warning:

NAVAID NAME MISSING

This problem is not limited to entering the data, it applies equally to deleting data as discussed below in Section b.

b. Deleting a Navaid

The keyword to be used for deleting a navaid entry from the data set is XNAVAI. Upon recognizing this keyword the program will print:

+++ EXPUNGE NAVAID

The code for the navaid to be deleted is placed in the first alpha field (column 71-74) of the card. Only one navaid may be deleted per card. If the program cannot find the entry, the program will issue the warning:

ENTRY **** NOT KNOWN

c. Purging the Navaid Data Set

Noise and performance data is keyed to aircraft and does not vary from airbase to airbase. Navaids are fixed at a particular ground location. Although a given navaid may be in use for SIDs of several air bases, there will in general be different navaids for different airbases.

If several airbases are run in sequence, and they use the same navaids then one can keep their location in the machine provided that the geometrical information for all bases is scaled from the same map coordinate system for all bases. If this is not the case or bases with different navaids are used, the proper navaids with the proper coordinate must be entered. If the navaid table in the program is full or approaching capacity, one may wish to delete non-current navaids. Since often all navaids are to be replaced and new ones entered and since the XNAVAI card only expunges a single entry per card, it is expeditious to use a new keyword to purge the

entire navaid directory. The keyword is CNAVAI and when recognized by the program it produces the message:

```
+++  PURGE NAVAID DIRECTORY
     ** ENTRIES DELETED

          0 ENTRIES(S)  25 CELL(S) AVAILABLE
```

d. Listing the Navaid Entries

A complete listing of all entries in the navaid data set may be obtained by using the keyword LNAVAI. Upon regognizing this keyword, the program will print the following:

```
+++  LIST ALL NAVIGATIONAL AIDS
```

A complete listing of all navaids in the data set will be listed. This listing includes the alphanumeric code and the (x,y) coordinate pair for each entry. If a FLIGHT card followed a DEPART card which referenced a navaid which was not present in the data set the code characters used on the DEPART card will be listed followed by the words "forced entry" as in:

```
          SMA            FORCED ENTRY
```

The listing of navigational aids may on occasion be initiated by the program. Since it is not due to a card, the card identifier '+++" will not appear in the margin. This listing is printed when the navaid table is full as an added diagnostic to expedite deleting unnecessary entries.

SECTION V

SEQUENCE DEPENDENT DATA CARDS

1. CODING AIR BASE OPERATIONS

The control cards and data cards which we have thus far discussed have been sequence independent according to the definition of sequence dependence given earlier. Aircraft operations are communicated through sequence dependent cards. That this is so should not be too surprising: one can describe each aircraft movement as an ordered sequence of events.

Allowable sequences of cards are completely specified in section II.4 using a formal metalanguage. The purpose of this section is to discuss in some depth the semantics of the sequence dependent cards. Although the cards discussed are given in sequential order the order chosen is usually not unique, and shows merely an example of the rules of section II.4.

To show the many different features of the sequence dependent cards and some of the precautions one must take under special conditions the following discussion centers around a hypothetical airfield: Falcon Air Force Base. This example base has some highly unusual regulations concerning runway use, and is equipped with an unlikely complement of aircraft in order that most typical situations can be discussed using only one example.

2. SEQUENCE DEPENDENCE

We first must reconsider the requirements of sequence independence before going further with the discussion of operational data. For a card to be sequence independent it must have two

properties:

1. It must always be recognized without an error message indicating that it was out of sequence.

2. The insertion of the card at an arbitrary point should not disturb the meaning of any sequence of sequence dependent cards.

There are three cards which one might call pseudo sequence independent. These three cards (AIRFLD, RUNWAY, RNPPAD) satisfy the first criterion.* They do not meet the second one, however. These three cards form the conclusion of one sequence and the beginning of the next. (There is a similar problem with one control card: all control cards are sequence independent except, strictly speaking, the END card, since all cards following it will be ignored by the program.)

The sequence dependence of the operational information is possibly best illustrated by the following bottom-up discussion of the data input. The card which ultimately causes the computation of the NEF values is the FLIGHT card. This card describes the type of aircraft, the type of mission, and the number of operations taking place. There must be a card which precedes this card to indicate where exactly this operation is taking place. In other words the aircraft flight path must be communicated.

The flight path is communicated effectively on two different cards. The altitude profile for the aircraft is part of the performance data package, and if not a standard default of the program, must be given on an ALTUDE card. This card is sequence

*Strictly speaking the RUNWAY and RNPPAD cards will only cause no error if at least one AIRFLD card precedes them.

independent according to the definition, but must obviously be present somewhere ahead of the point where it is needed. The projection of the flight path onto the ground plane is communicated on the FLTTRK card which describes the flight track explicitly, or may be communicated as a departure procedure using a DEPART card (provided that the operation is a takeoff). The only sequence dependent card which may be present between a FLTTRK (or DEPART) card and a FLIGHT card is another FLIGHT card.

It is also clear that a flight track must either begin or end on a runway. Therefore ahead of a FLTTRK or DEPART card there must be a RUNWAY card to describe the location of the runway. The RUNWAY card is the beginning of a card sequence of sequence dependent cards. Since a RUNWAY card must precede all explicit or implicit definitions of flight tracks one must know which intervening sequence dependent cards are allowed. Since many flight tracks may originate from a single runway a FLTTRK or DEPART card must be preceded by a RUNWAY card, or by any number of FLTTRK cards followed by any number of FLIGHT cards, provided that the first of these FLTTRK sequences is preceded by no other sequence dependent cards than a RUNWAY card. In the above one may substitute DEPART for FLTTRK at will.

One may appreciate from the above description why in section II.6 the BNF metalanguage was introduced to keep the description of card sequences tractable. As mentioned before the restrictions presented above apply only to sequence dependent cards. There is no exception to the rule that any number of sequence independent cards may be interspersed anywhere in the deck structure.

One important thing to remember is that whenever a RUNWAY card is read the previous runway is no longer accessible to the program. Similarly a DEPART or FLTTRK card will replace the information of the previous flight track and one must re-enter the pertinent information when one wants to go back to a previous runway or flight

track. It is most efficient to complete all operations for a given flight track before going on to the next, and to complete all flight tracks before going on to the next runway.

A similar argument can be made for runup operations. The runup itself is described by a RUNUP card which describes the duration of each runup, the number of runups, the aircraft type and the power setting. Runups take place at positions on an airfield called runup pads which are defined to the program by giving their location on a RNPPAD card. The only sequence dependent cards allowed between a RNPPAD and a RUNUP card is another RUNUP card.

Although the noise and performance data are input as sequence independent information one must of course always have entered the necessary data before a FLIGHT or RUNUP card is encountered. If no information is present the program will issue an error. If the wrong information is present no diagnoistic whatsoever is provided, since only the user can decide which information is appropriate in a given situation.

3. FLIGHT OPERATIONS

a. Starting a New Airbase

A new airfield is started when an AIRFLD card is encountered. The program will conclude the previous airbase and start the next one. If the program is in the initialization phase the necessary steps will be taken to conclude this phase. In either case an error summary is printed. The AIRFLD card forms the start of a new sequence which will continue until the next AIRFLD or an END card is encountered.

b. The Error Summary

At the conclusion of an airfield the program prints a summary of error and warning messages. This summary consists of a listing of the page numbers on which errors and warnings occurred during this airfield. If none occurred in a category the word NONE will be printed. The program can keep track of up to 56 pages with errors and up to 200 pages with warnings. An example of a typical error summary is shown in Figure V.1

```
  FALCON AIR FORCE BASE - SUNNYVALLEY, CALIFORNIA              PAGE  176
------------------------------------------------------------------------

ERROR STATISTICS
================

            FATAL ERRORS -      OCCUR ON PAGE(S)
                                    8   78  166


            WARNING MESSAGES -  OCCUR ON PAGE(S)
                                   14   66   89  124  172
```

FIGURE V-1. TYPICAL ERROR SUMMARY

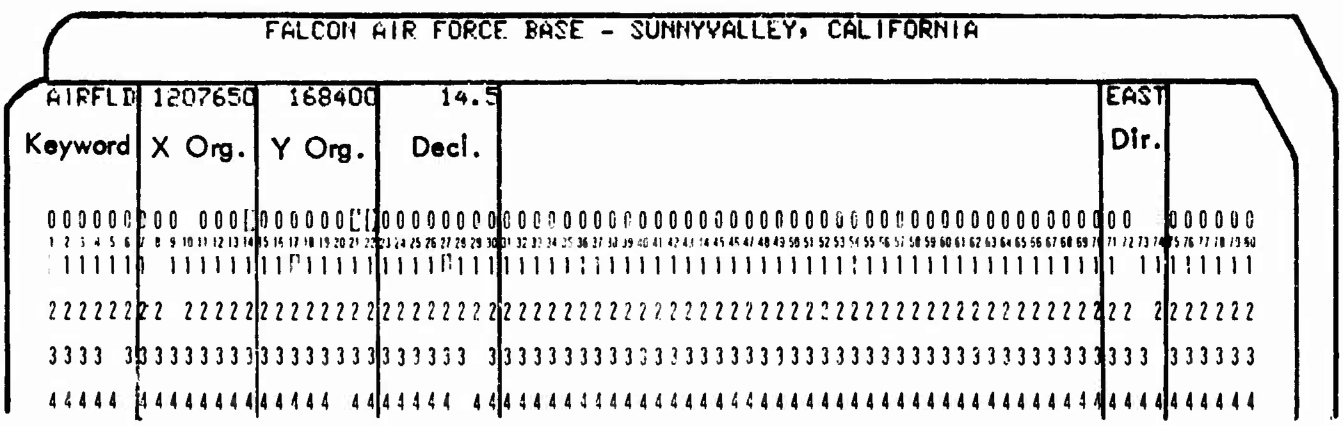
If the program found errors or warnings on more pages than it could keep track of, the error count beyond the last page number which could be stored will be printed as signified by the following message appended to the listing:

```
AND IN **** MORE INSTANCES AFTER PAGE ****
```

Errors which occurred during the initialization procedure cause the program to stop execution with the message:

```
EXECUTION TERMINATED DUE TO INITIALIZATION ERRORS
```

c. The AIRFLD Card




```
+++  NEW AIRFIELD FALCON AIR FORCE BASE - SUNNYVALLEY, CALIFORNIA
                    EXTERNAL LOCATION OF GRID ORIGIN X = 1207650 Y = 168400
                    MAGNETIC DECLINATION  14.5 DEG TO EAST
                    OPTIONS  PROGRAM WILL ANALYZE INPUT DATA  (ENGLISH UNITS)
                             BUT NO PROCESSING WILL BE DONE
                             DATA BASE WAS RESET
                    FILES KNOWN TO PROGRAM
                    UNIT 10 BINARY WITH    0 DUMPS
```

The AIRFLD card is unique in that it actually consists of two cards. The second card contains any suitable alphanumeric descriptor which will be printed as the page heading for all output of the program.* The AIRFLD card is physically two cards and no continuation character is needed in the continuation field of the first card.

The AIRFLD card will cause the grid to be cleared irrespective of the mode of the program and will reset the data base to its initialization values unless this is inhibited by a NODATA card. The options list gives the options in effect when the AIRFLD card was read. These may, of course, be changed at any time.

The following discussion uses FALCON AFB as an example. This airbase is fictitious. The aircraft complement is assumed to be F-100 fighter aircraft, C-141 cargo aircraft and B-52 bomber aircraft. This heterogeneous group is flown following a set of equally unusual flight paths and departure routes. The only reason is to present the cards in the order which one would ordinarily use them for a "real" case, but at the same time show how all different parameters are used on the cards. Appendix A gives the chronicle listing of a simple test case to show some of the features discussed.

1) Grid Origin

The computer grid must be associated with the reference used to input the data which are to follow. One should therefore decide where to put the grid for which NEF values will be computed. A first step in the absence of any contrary judgment is to positon the grid so that the airfield is in the middle. The grid origin is located at the lower left hand corner of the grid.

*This descriptor should be placed in columns 7-66.

It is important to realize that the grid must be aligned with the coordinate system used to scale distances. This would be most conveniently the coordinate system of the Coast and Geodetic Survey (C.&G.S) map. When the area for which NEF contours must be computed is defined one can then locate the lower left hand corner of the square on the C.&G.S. map. The coordinates of this point should be entered in the X-ORIG, Y-ORIG fields of the AIRFLD card.

2) Magnetic Declination

The program is capable of constructing flight tracks from departure procedures. These procedures are worded in terms of magnetic heading. All geometric calculations in the program are done with respect to the NEF grid. It is therefore necessary to specify the magnetic declination with respect to Y-axis of the grid. The number of degrees should be entered on the DECL field, and EAST or WEST coded in the DIR field.

The magnetic declination specified must be in the range $0 \leq DECL \leq 180$ and the word EAST or WEST must be placed in column 71-74. If these conditions are not met the program will issue the error:

ILLEGAL MAGNETIC DECLINATION ***.* DEG TO ****

It is allowable not to specify a declination. In that case all headings are in degrees true, but care must then be taken that no magnetic headings are used for input.

3) Airfield Title

The AIRFLD statement consists of two cards. The second of these, which must always be present, is used to communicate an airfield title. This title will be

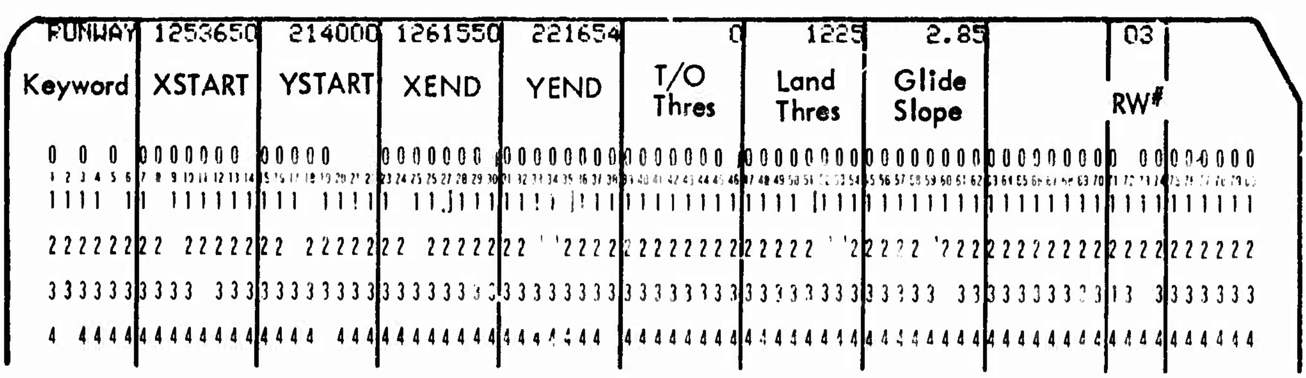
printed on the chronicle pages and separators. It is also written on the GPCP plots and is part of the header information written on binary tapes.

4) Allowable Sequences (Functional Relationship)

If we compare NEFUSAF statements to natural language one may consider a computer run as a book. The initialization phase forms the preface. Each AIRFLD starts a new chapter. Each chapter is divided into several sections by RUNWAY and RNPPAD cards. Each FLIGHT card starts a new paragraph in the section.

Now that we have located our grid on the map we can use the coordinates of the map directly to scale off our further input. The first thing to do is to define a runway. The major runway at Falcon AFB is 03-21 so we will start with this.

d. The RUNWAY Card




```
+++                   ---------------------------
                          R U N W A Y     03
                      ---------------------------

                 LENGTH 11000 FT, GLIDE SLOPE  2.85 DEG, HEADING 32 DEG
                 START(1253650,  214000)  END(1261550, 221654)
                 DISPLACEMENTS - TAKEOFF       0, LANDING    1225
```

The RUNWAY card starts a sequenced group of cards describing the operations on that runway. To define the runway we define its centerline. The beginning and end of the runway centerline is carefully measured from the map and noted. Since the runway can be operated as either 03 or 21 we must first define the directionality.

1) Directionality of Runway Operations

Operation of the runway as 03 implies that the aircraft operating on it fly toward the north, on runway 21 they would fly toward the south. This is therefore the way in which we define the directionality. For runway 03 the south end should be coded in the XSTART, YSTART fields. The north end will then be coded in the XEND, YEND fields. For runway 21 the opposite applies.

2) Admissible Runway Length

The values coded in the XSTART, YSTART, XEND and YEND fields are centerline coordinates, defining runway length as well as orientation. If a runway is longer than 16000 feet (4876m) a diagnostic will be printed in the chronicle:

RUNWAY LENGTH IS GREATER THAN 16000 FT

This is a warning to indicate that an unusually long runway was encountered, which may indicate a keypunch error.

3) Displaced Threshold

In many cases there are displaced thresholds. Most often these occur for landings but they may equally well occur for takeoffs. (Intersection takeoffs are in this

category.) For displaced thresholds the amount of displacement should be entered on the appropriate fields on the card. On runway 03 the displacement is 1225 feet, leaving 9775 feet of available runway. (The aircraft is generally assumed at 50 feet AGL over the threshold for landing operations.) The displaced threshold for landings is shown on a plot as a bar across the runway. It should be emphasized that when a displaced threshold is given <u>all landings without exception</u> will use this threshold. (See page 115, Displaced Threshold not Used by All Aircraft.) Takeoff threshold are implemented similarly, except that it applies to departures and that no threshold location is shown on the map.

The following errors may occur:

TAKEOFF DISPLACEMENT IS ILLEGAL

LANDING DISPLACEMENT IS ILLEGAL

The displacement in that case is either larger than the runway length or it is negative.

4) Glide Slope

The program has the capability of generating a landing profile from a glide slope. The program will assume an aircraft altitude of 50 feet AGL over the (displaced) landing threshold. If no value is given or the value 0 is coded, the program will assume a 3° slope. Legal slopes are contained in the interval $0.5° \leq \text{SLOPE} \leq 10°$. Any other value results in the error:

ILLEGAL GLIDE SLOPE

If a steeper or shallower approach is desired the user will have to specify such an altitude profile explicitly. That is, he will have to compute the necessary altitude-distance curve and enter it on an ALTUDE card.

5) Runway Number

The runway number may be entered in the RW # field of the card. This information will be printed in the chronicle. It will also be put on the runway when a plot is made which includes a flight track map. On the plot the four characters of the field will be centered, so that the left two characters will be to the left of the runway centerline, the remaining two to the right. The number will appear in the "clear zone" for the runway. It will be in a direction such that an aircraft landing on the runway will see it right side up.

6) Runway Width

All runways will always be plotted as a 250 foot wide strip, irrespective of the actual width of the runway. Since the width is of no importance in the NEF calculation, no provision is made to enter this item into the computer.

7) Inactive Runways

The next step after defining a runway and its direction of operation, is to define the operations on this runway. If the runway is closed no operations take place and no further cards are required. If the runway is used but the operations on this runway are not considered for inclusion in an NEF map, we may also omit any further

cards. It is clear that we could have omitted the RUNWAY card altogether in these cases. By putting the card in we will, however, get a complete runway layout on any subsequent plot when we ask for a runway/flight track map.

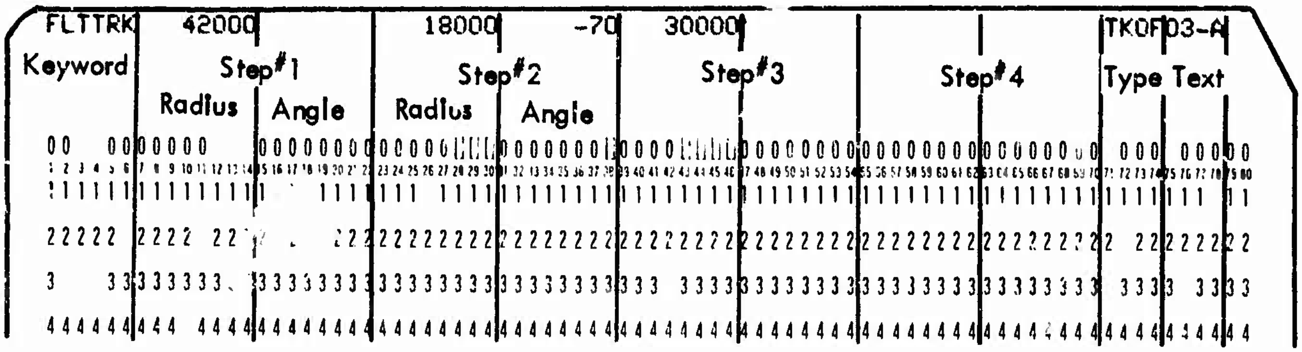
8) Non-Empty Sequences After a RUNWAY Card

The RUNWAY card concludes any previous sequence and initiates a new one. If the RUNWAY is not inactive a FLTTRK or a DEPART card is the next card. A FLTTRK or DEPART card describes the flight track which aircraft follow while airborne. Each FLTTRK or DEPART card can be followed by as many FLIGHT cards as are appropriate to this particular flight track or procedure. When all FLIGHT cards for a particular flight track have been read the next flight track or procedure for this runway may be entered. This card is then followed by as many FLIGHT cards as necessary. The process is entirely recursive; any number of legal sequences of FLTTRK, DEPART and FLIGHT cards constitutes a legal sequence.

e. Lewis Three Departure Runway 03

Large aircraft generally use the Lewis Three departure (Figure VI.7). The departure via the Molehill transition is used by flight training missions. B-52 and C-141 aircraft will practice T & G at the Lake Louise base. The cargo hauling flights generally follow the Westchester Transition. The bombing practice flights proceed over Merry Island. Departure procedures are discussed in Chapter VI, and the Lewis Three Departure is discussed at the end of that chapter.

f. The FLTTRK Card




```
+++  TAKE-OFFS        FLIGHT TRACK     03-A
                      PROCEED         42000 FT
                      TURN LEFT          70 DEG WITH 18000 FT RADIUS
                      PROCEED         30000 FT
```

1) F-100 T/O Runway 03

Takeoffs of F-100 aircraft on Runway 03 proceed along radar vectors. In the vicinity of the airport the procedure is to keep the aircraft on runway heading until the outer marker of Runway 21. Aircraft then turn to a heading of 320° and proceed outbound for several miles before being assigned further headings. The FLTTRK card is used to describe a flight track explicitly.

2) Straight or Curved Tracks

A straight flight track is entered by coding the length of the straight line segment in the RADIUS field. The ANGLE field is to be left blank. The first segment specified must be a straight line section. The error

FLIGHT TRACK DOES NOT START WITH A LINE SEGMENT

will be printed in the chronicle if this is not done.

A curved track requires an entry in ANGLE as well as the RADIUS field. The RADIUS field describes the radius of the turn. The ANGLE field describes the angle over which the turn must be made. Righthand turns are coded by using a positive value for the angle; a lefthand turn requires a negative entry in the ANGLE field. If the angles specified exceeds 360 degrees a warning will be printed:

```
ANGLE SPECIFIED ON FLTTRK CARD IS GREATER THAN 360 DEGREES
```

3) Type of Operation

The flight track on a FLTTRK card is either a takeoff or a landing. The program must be told which type a given track is. This is accomplished by entering in columns 71-74 the four characters TKOF for a takeoff or LAND for a landing. To correlate a printout early with a flight track map one may put some identifier in columns 75-78. This identifier will print in the Chronicle but it will not appear on any plot generated by the program.

4) Complexity

The program allows a user to input up to 25 segments in a flight track. If more complex flight tracks are entered the error:

```
TOO MANY SEGMENTS IN FLIGHT TRACK
```

will be printed in the chronicle. If this happens one should carefully check his flight track and delete unnecessary complexities.

5) Continuation

Up to four segments may be coded on a single card. The program will scan cards from left to right and stop on a field with both RADIUS and ANGLE equal 0. If a continuation character is present it will be recognized and the next card read. This will occur whether or not all four fields on the previous card were used. The program will check for certain minimum conditions to be satisfied. The flight track must start with a straight section or the error

FLIGHT TRACK DOES NOT START WITH A LINE SEGMENT

will be printed. Similarly, if angles are used they will generate, if their absolute value is more than 360 degrees

ANGLE SPECIFIED ON FLTTRK CARD IS GREATER THAN 360 DEGREES

The maximum number of flight track entries is 25. If more are entered the program will print the error

TOO MANY SEGMENTS IN FLIGHT TRACK

The program will keep reading continuation cards and list entries even after the storage capacity is exceeded. A very similar message may be generated after a FLIGHT card:

TOO MANY SEGMENTS IN FLIGHT PATH

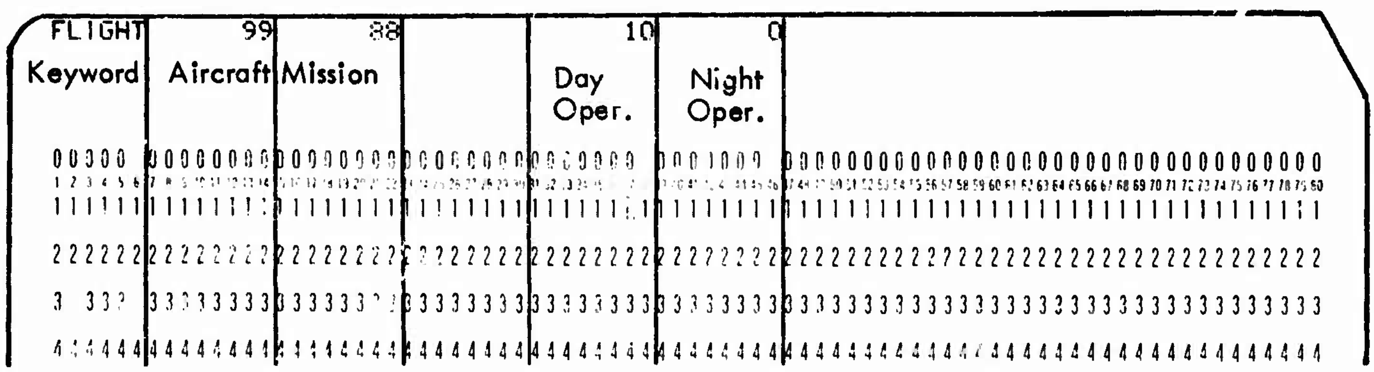
This message means that the internal working storage for the program in which the information for all altitude, delta EPNL profiles and subflights is kept is exceeded. The combined complexity for all information for this FLIGHT/FLTTRK combination is too large to be handled.

A continuation card without a blank keyword field will result in the error

```
ILLEGAL CONTINUATION AFTER 'FLTTRK' CARD
```

g. The FLIGHT card

The FLTTRK card is now followed by the FLIGHT card for the F-100 operations.




```
+++  F-100       AIRCRAFT NO. =      99      MISSION NO. =  88
                 OPERATIONS - DAY  10.000 , NIGHT       .000
```

The first data field contains the aircraft number, the second data field the mission number. The card will cause the flight descriptor data set to be searched for this particular aircraft/mission number. If the combination is found the associated data are made accessible to the program and the text descriptor (F-100) will be printed. If the entry does not exist the program will instead print

```
INVALID
```

Data field 3 is not used and should be left blank. Data field 4 contains the number of daytime operations of this type of aircraft on the flight track specified in the last FLTTRK card. Data field 5 contains similarly the nighttime operations.

The total number of operations specified on the card must be larger than zero. Neither daytime nor nighttime operations may, of course, be negative. In any of these events the following warning appears:

WEIGHTED OPERATIONS ***.*** ILLEGAL NO COMPUTATION FOR THIS FLIGHT

The program may issue the following mostly self-explanatory warnings and errors:

AIRCRAFT NUMBER SPECIFIED ON THE FLIGHT CARD IS NOT PRESENT IN THE DIRECTORY

ALTITUDE PROFILE NO. ***** SPECIFIED IN FLIGHT DESCRIPTOR HAS NOT BEEN ENTERED

POWER LEVEL PROFILE NO. ***** SPECIFIED IN FLIGHT DESCRIPTOR HAS NOT BEEN ENTERED

INTEGRATED NOISE LEVEL PROFILE NO. ***** SPECIFIED IN FLIGHT DESCRIPTOR HAS NOT BEEN ENTERED

AIRCRAFT IS STILL ON THE GROUND AT THE END OF THE RUNWAY

DISTANCE COVERED BY SUBFLIGHTS IS LESS THAN THE TOTAL FLIGHT TRACK

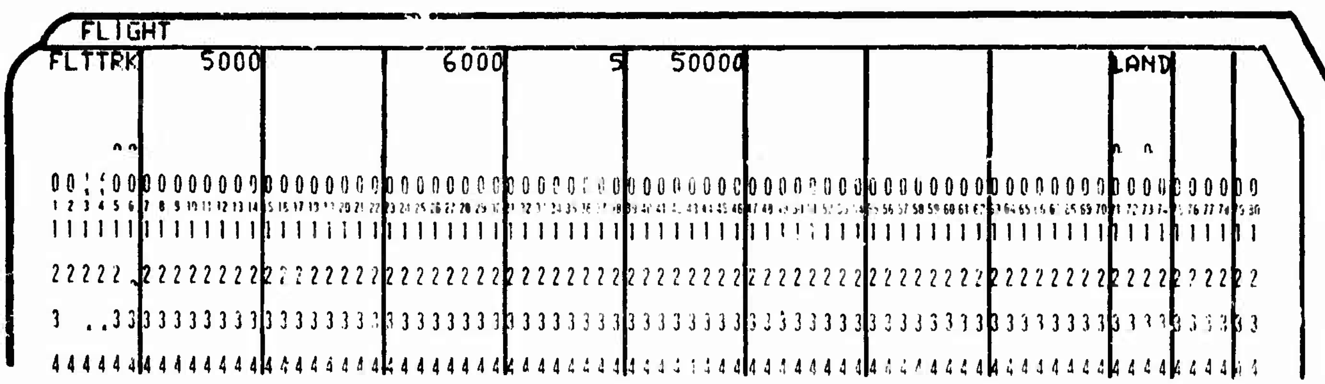
AIRCRAFT IS NOT AIRBORNE AT THE START OF TURN

In the above the "power level profile" is a Delta EPNL profile, the "integrated noise level profile" an EPNdB profile. In the last error the implication is that the turn is the first one in the flight track.

h. Some Further Considerations for Flight Operations

1) GCA/ILS Runway 03

Approaches to Runway 03 are either Ground Controlled Approaches or ILS approaches. In either event aircraft will proceed on the 2.85° glide slope from the outer marker inward. The card sequence is:



2) Runway 21

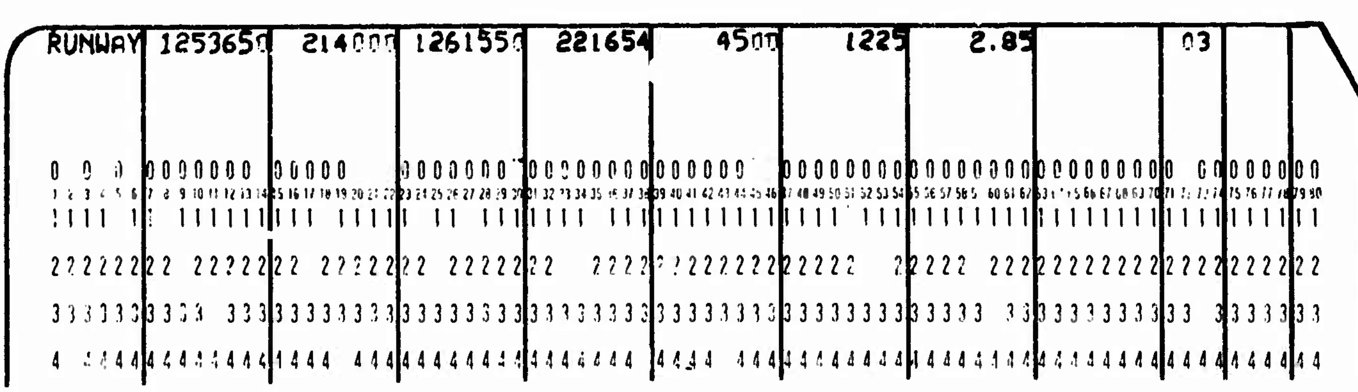
Sometimes Runway 21 will be used. The card is of course almost identical to the card for Runway 03, except that the ends are interchanged. The label is now 21, and there is no threshold displacement for landings. Small fighter aircraft (F-100) will make intersection takeoffs. We will consider those operations which use the full runway length first. Since for these aircraft there is no displaced T/O threshold the T/O DISPL field is left blank on the card.

3) Displaced T/O Threshold

High performance aircraft will often make an intersection takeoff on Runway 21. Since not all aircraft do this we have first considered the aircraft which use the full length of the runway. The intersection takeoff is equivalent with a displaced takeoff threshold: the start of the takeoff roll is not at the physical end of the runway. A displaced threshold on a RUNWAY card will cause all takeoffs to start from that point. If aircraft routinely back up into an overrun area before takeoff, the end of the overrun area becomes effectively the end of the runway and this point should be coded on the RUNWAY card since negative thresholds are not allowed. Since landings will use the normal threshold a landing displacement equal to the length of the overrun area must then be included.

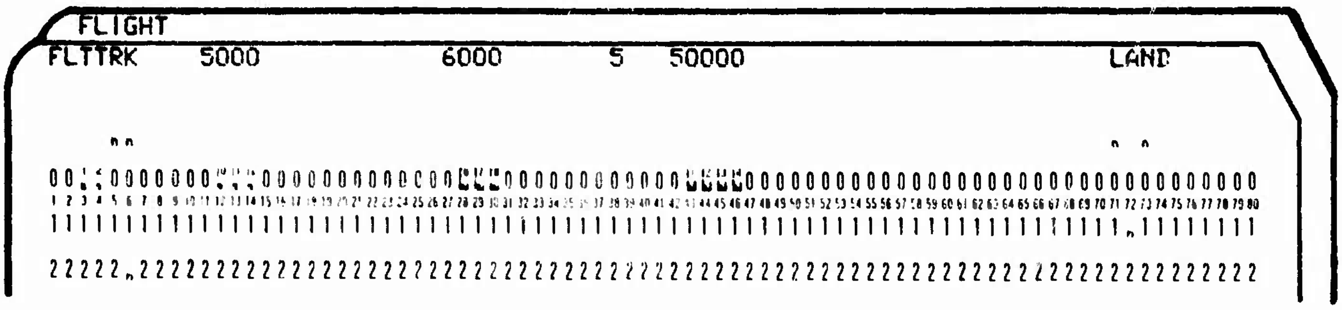
4) Displaced Threshold Not Used by All Aircraft

In our current example only some aircraft (F-100) will make intersection takeoffs. We therefore need an additional RUNWAY card to describe these operations. When only certain types of aircraft use a displaced threshold the runway with displaced threshold and the runway without displaced threshold should be considered entirely separate. Although the two RUNWAY cards describe operations on the same physical runway they are logically distinct to the program. We therefore need an additional set of cards to describe these operations.



5) TACAN Approach Runway 21

The F-100 aircraft will generally make a TACAN approach to Runway 21. The 035 radial intersects the runway center line 5000 feet from threshold. Since the runway magnetic heading is 30° the turn to final is 5°. The actual path flown can be described as:



6) More Complex Airbases

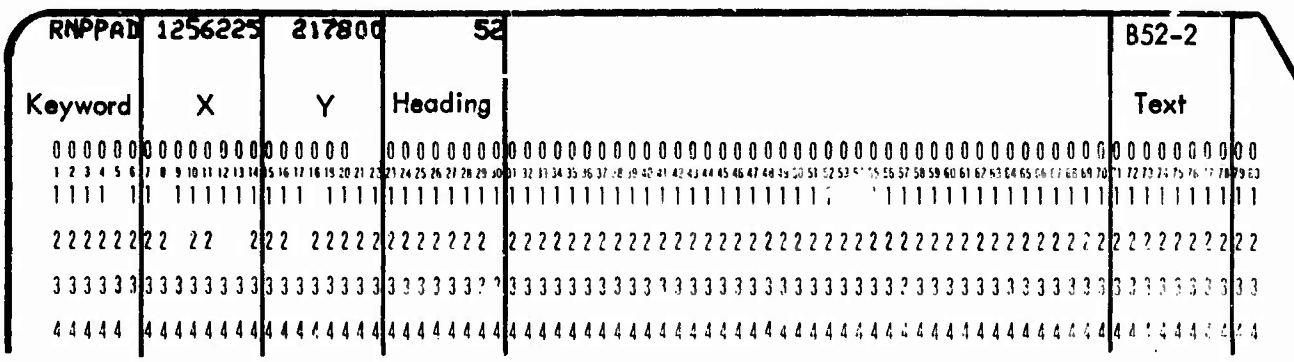
The airbase described above is fairly simple. More complex airfields are, however, basically a superposition of many simple bases. Any airbase can be described by steps as outlined above. The examples above give a fairly complete cross-section of military jet operations as one would encounter them on real airbases.

In general the type of aircraft at a base will be more homogeneous, but the number of different flight paths may be larger. Each possible path can be coded in terms of the above cards. Whether one uses a FLTTRK or DEPART card is up to the user. It depends entirely on which of the two methods will most correctly reflect the operations. At times both methods will be entirely equivalent, in which case the user's preference is the only factor.

4. GROUND RUNUP OPERATIONS

Ground runups are also described by a sequence of sequence dependent cards. The sequence is initiated by a RNPPAD card followed by as many RUNUP cards as necessary to describe the runups at the runup pad given on the RNPPAD card.

To define a runup pad one uses the keyword RNPPAD.

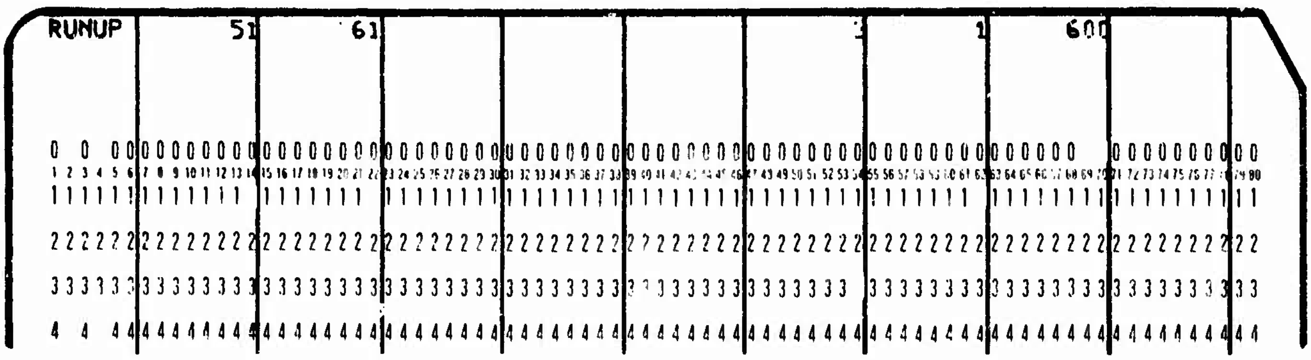



```
                    --------------------------------
+++                   R U N U P    P A D  B52-2
                    --------------------------------

X =    1256225 FT, Y =    217800 FT. HEADING =  52.0 DEG.
```

The first and second data field contain the X and Y coordinate of the runup pad. The third data field contains the aircraft heading in degrees magnetic assumed for runups on this pad. The text field may be used for any identification.

The actual runup time for aircraft is communicated on the RUNUP card.




```
                                        RUNUPS PER TIME PERIOD     DURATION OF
+++   AIRCRAFT CLASS        THRUST      0700-2200   2200-0700      EACH RUNUP
         51 B-52              61            3           1           600 SEC
```

The first data field contains the aircraft number and the second field the thrust number. The information will be compared to the data set of runup discriptors and the program will make the necessary information available. If the aircraft/thrust combination is legal, the program will print the alphanumeric identification, if any, if the entry does not exist, the program will print

INVALID

instead. The first time a particular invalid entry occurs the program will print the error message

RUNUP DESCR. FOR THIS COMBIN. MISSING

Other possible errors may be:

MAX. NOISE LEV. PROF. **** MISSING

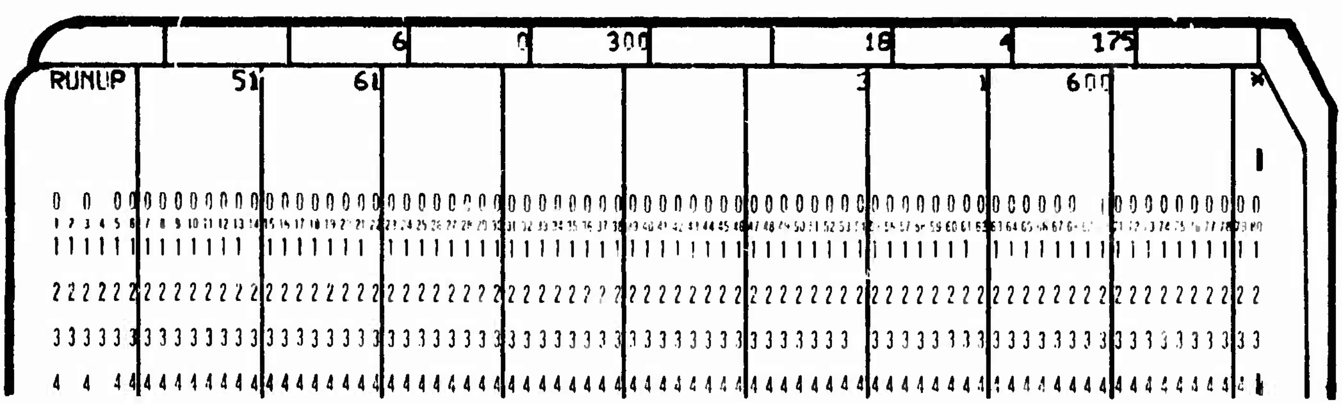
PAD IN USE MORE THAN 15 HRS/DAY

PAD IN USE MORE THAN 9 HRS/NIGHT

These last two errors indicate that the total runup time for all aircraft on this pad has exceeded the number of seconds in the daytime or nighttime period.

The number of daytime runups is placed in data field 6, the number of nighttime runups in data field 7 and the duration per event in field 8. If aircraft are run up with different run times it is much better to use continuation cards. When a RUNUP card is used the program will start calculating NEF values on the first runup before looking at the second. If a continuation card is used the program will calculate a cumulative runup time and compute NEF values only once. If significant changes in thrust settings are used this is not useful, of course. In the above example it is much better to use a continuation card for the runups of 300 and 175 second duration which also take place for thrust entry 61.

The continuation card will have the daytime and nighttime number of events in data fields 2 and 3 respectively. The fourth data field contains the duration of each event (300 seconds). The third runup time (175 seconds) can be entered similarly in fields 6, 7 and 8.



SECTION VI

DEPARTURE PROCEDURES

1. THE DEPARTURE PROCEDURE CONCEPT

The most powerful method of entering flight track information into the program is through the use of a departure procedure. The departure procedure as entered on a DEPART card is logically equivalent with entering a flight track on a FLTTRK card. The result is very different, however, since the DEPART card does not generate a flight track at the time of entering, but only "compiles" the information. The explicit definition by means of a FLTTRK card will cause all subsequent aircraft referenced in a FLIGHT card to follow exactly the same ground track. The implicit definition by means of a DEPART card will cause all subsequent aircraft referenced in a FLIGHT card to generate the flight track appropriate to the performance characteristics of each articular aircraft.

Each aircraft/mission combination has associated with it a certain altitude profile and a turn radius. When a pilot's instruction is to climb to a given altitude and then turn to a certain heading,*the flight track generated by these instructions will vary with the performance characteristics of the aircraft flown. The DEPART card allows the user to specify only the clearance received by the pilot. The program will then generate the corresponding flight track.

*The program assumes all headings and radials to be in degrees magnetic.

It is also possible to include altitude restrictions in such a procedure. If an aircraft is to stay below a certain altitude until a specific point, this may be included in the description of the procedure. The program will then modify the altitude profile specified for the non-restricted performance and insert such portions of level flight as may be necessary to enforce the restriction. For this part of the program to work satisfactorily, one should not start with an altitude profile which already has level flight segments in it, since the resulting altitude profile is unpredictable.

The increased data entry capability is offset by a decrease in diagnostic efficiency. As was pointed out earlier, NEFUSAF is capable of detecting all syntactic errors in one pass, but requires several passes for semantic errors. Since the DEPART is a highly contextual statement, the diagnostic capability per pass is limited. That this is so can easily be seen. If an error occurs, the program cannot check the remainder of the procedure since one cannot lay out the flight track beyond the point where the error occurred. Similarly, if a warning was issued because the program found that some choices were nonsensical, the correction of that problem will change the context in which the remainder of the procedure must be interpreted.

It bears repetition at this point to review the meaning of ERROR and WARNING as issued by the program. An error is issued by the program when the program cannot proceed with the calculations. As such it points to an error in the data input. A warning is issued by the program when the program detects a condition where the likelihood of errors in the input is considerably higher than usual. The program cannot decide whether or not the data were in error and processing is still possible. One must, however, not forget that the qualification "higher than usual" implies that it is possible, at any time, to give erroneous information to the program which will go undetected! An

error summary showing no errors and no warnings only conveys to the user the information that the data cards were syntactically and semantically correct statements of the NEFUSAF language. No diagnostic is provided as to whether the data correctly describes runway layouts, flight paths, noise and performance data or operations of the air base under study.

The execution of a departure procedure is signified by the message, FOLLOWS DEPARTURE PROCEDURE, following the alphabetic descriptor of the aircraft as printed in the Chronicle when a FLIGHT card is recognized. (If an error or warning message was printed as a result of the FLIGHT card, the message will appear underneath the last message banner.) This entry is then followed by the diagnostics, if any, generated during the departure procedure execution. After completion of the procedure, the altitude/distance curve is printed, followed by a listing of maneuvers executed. If during the "compilation" of the procedure, an error was detected which would cause the execution of the procedure to be erroneous or impossible, the program will only print the message:

```
EXECUTION OF PROCEDURE SKIPPED DUE TO PREVIOUS ERROR(S)
```

This error may also be caused by a missing altitude profile, which is always considered an error, even when no reference to an altitude is made in the wording of the procedure. One should note that a missing navaid does not give this message. Therefore, although the "skipped execution" message appears immediately after certain navaid errors, the reason for the skipping is never due to missing navaids. Errors due to navaids and their Chronicle entries are further discussed in Section IV.

Errors may be detected at two different points. Some errors will be found when the DEPART card is read, others will not be found until the subsequent FLIGHT card is read. The errors

which are deferred to the FLIGHT card are not necessarily undetectable at an earlier stage. It is more convenient and more efficient to check the item at a later time, and better diagnostics can then often be provided. An example is shown below:

```
CLIMB TO      1500 FT THEN
TURN TO HEADING  270

PROCEED DIRECT TO NAVAID NNV
   RESTRICTIONS
   FOR NEXT  46000 FT
   STAY BELOW 1000 FT
```

It is a rather glaring error to enter a restriction to 1000 ft. in the second step after first having allowed the aircraft to climb to 1500 ft. in the first. Nonetheless, the program will not detect this error until after the FLIGHT card has been read. At the point where all other illegal altitude restrictions will also be detected, the error message will be generated:

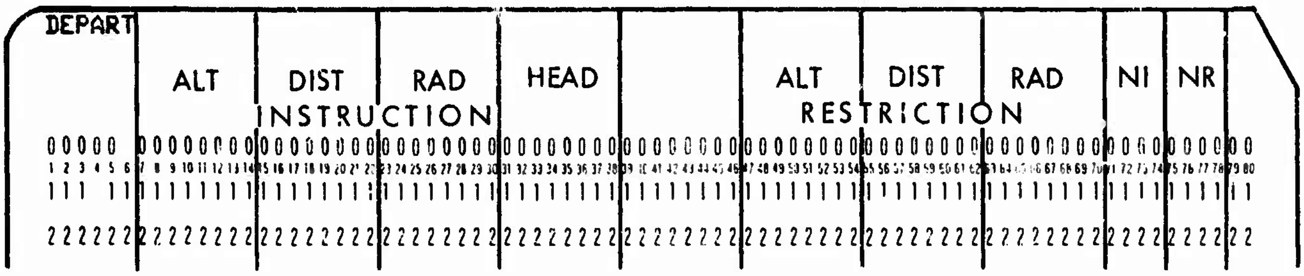
```
ALT. REST.  1000 FT ILLEGAL AFTER  25477 FT WHEN A/C IS AT 1687 FT.
```

indicating that after reaching 1500 ft. the aircraft has turned to the assigned heading of 270 (during which maneuver it gained an additional 187 ft. of altitude). Then, when the next instruction step was encountered (proceed direct to navaid NNV), the restriction to 1000 ft. was found which was at that moment clearly illegal. The flight track which has been generated thus far will, however, be plotted if a PLOT card is encountered. The message following the above error will read:

```
PROCEDURE ABANDONED AFTER  1 STEP(S)
```

indicating that only the first step was executed and its flight track written on an internal file for later plotting. The current altitude profile and flight track will be printed after this message.

2. THE INSTRUCTION PART OF THE DEPART CARD



See Figure VI.1 for printout examples.

The first four columns of the card contain the instruction to be executed. The next three columns contain the (optional) restriction. The DEPART card itself must not contain a restriction only or the error message:

RESTRICTION PRECEDES FIRST INSTRUCTION

will be printed. Subsequent continuation cards may contain either or both types of entry. A card which has no data coded on it will be read and ignored. The total number of allowable continuation cards varies with the contents of the cards. Rules for this are given on page 142.

The ALT field in the instruction contains the altitude to which the aircraft must climb. The entry in the Chronicle will print as:

CLIMB TO ****** FT

The DIST field in the instruction contains the distance to be flown as measured from the current aircraft position.

+++ DEPARTURE PROCEDURE

PROCEED FOR 17000 FT THEN
TURN TO HEADING 150

INTERCEPT LLS 078 RAD (INBOUND)

INTERCEPT FLM 010 RAD (INBOUND)

PROCEED FOR 60000 FT

********** END OF PROCEDURE **********

+++ F-100 AIRCRAFT NO. = 99 MISSION NO. = 88
OPERATIONS -- DAY 10.000 , NIGHT 0.000

*************************** W A R N I N G ***************************
UNSPECIFIED TURN RADIUS RESET TO 6000 FT
****

*************************** W A R N I N G ***************************
A/C GONE BEYOND ALTITUDE PROFILE AFTER 3 STEPS
****

PROFILE GENERATED -- DISTANCE ALTITUDE

DISTANCE	ALTITUDE
0	0 FT
6000	0 FT
8000	100 FT
100000	19700 FT
115673	19700 FT

FLIGHT TRACK -- MANEUVER EXECUTED / DISTANCE AFTER COMPLETION / HEADING

MANEUVER EXECUTED		DISTANCE AFTER COMPLETION	HEADING
PROCEED	17000 FT	17000 FT	32 DEG
TURN RIGHT	119 DEG	29419 FT	150 DEG
PROCEED	54888 FT	84307 FT	150 DEG
TURN RIGHT	108 DEG	95617 FT	258 DEG
PROCEED	12935 FT	108552 FT	258 DEG
TURN LEFT	68 DEG	115673 FT	190 DEG
PROCEED	60000 FT	175673 FT	190 DEG

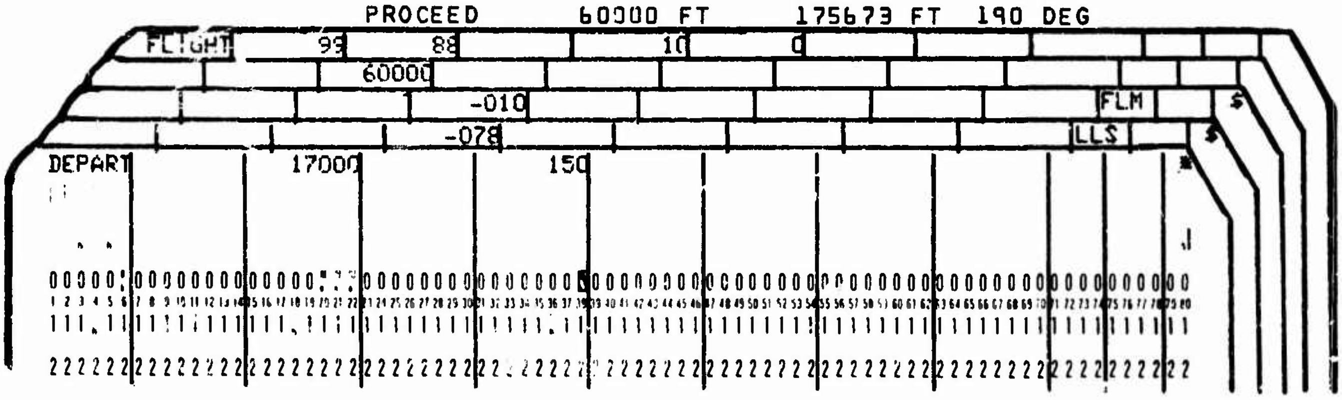


FIGURE VI.1 DEPARTURE PROCEDURE EXAMPLES

+++ DEPARTURE PROCEDURE

PROCEED FOR 17000 FT THEN
TURN TO HEADING 150
RESTRICTIONS
FOR 60000 FT
STAY BELOW 7500 FT

INTERCEPT FLM 010 RAD (INBOUND)

PROCEED DIRECT TO NAVAID FLM

*********** END OF PROCEDURE **********

+++ F-100 AIRCRAFT NO. = 99 MISSION NO. = 88
OPERATIONS - DAY 10.000 , NIGHT 0.000

PROFILE GENERATED -- DISTANCE ALTITUDE

DISTANCE	ALTITUDE
0	0 FT
6000	0 FT
8000	100 FT
42795	7500 FT
60000	7500 FT
117265	19700 FT

FLIGHT TRACK -- MANEUVER EXECUTED, DISTANCE AFTER COMPLETION, HEADING

MANEUVER EXECUTED		DISTANCE AFTER COMPLETION	HEADING
PROCEED	17000 FT	17000 FT	32 DEG
TURN RIGHT	119 DEG	29419 FT	150 DEG
PROCEED	24554 FT	53973 FT	150 DEG
TURN RIGHT	40 DEG	58162 FT	190 DEG
PROCEED	51619 FT	109780 FT	190 DEG

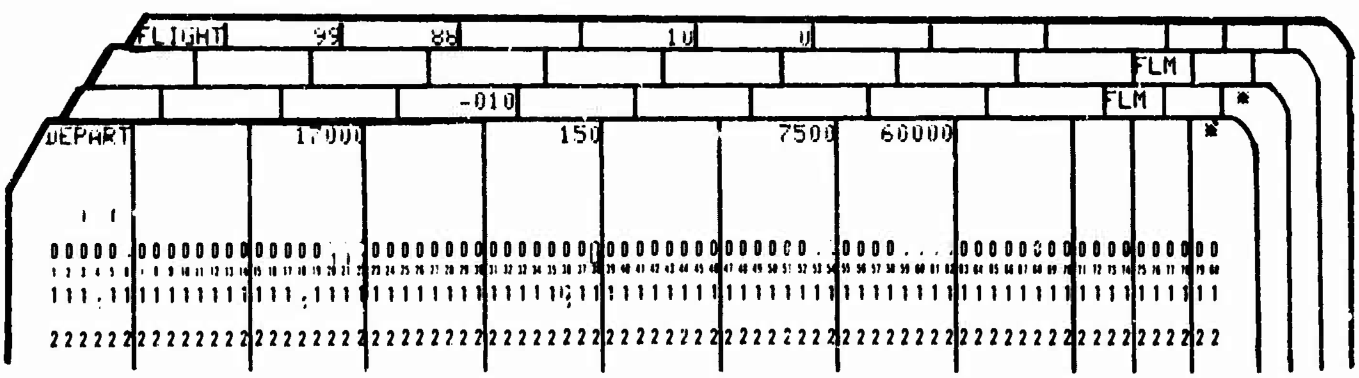


FIGURE VI.1 CONTINUED

The corresponding Chronicle entry is:

    PROCEED FOR ****** FT

The HEAD field contains an entry indicating the heading onto which the aircraft must turn. The Chronicle will echo:

    TURN TO HEADING ****** FT

The entry in the RAD field (in combination with the navaid name in NI) can print two different messages, depending on the context. If no heading was specified, the program assumes that the radial is to be intercepted. If the value is positive, the message is:

INTERCEPT **** ****** RAD

whereas wher the entry is negative, it is assumed that the following is intended:

INTERCEPT **** ****RAD (INBOUND)

The intercept instruction will move the aircraft from its current position onto the radial. No motion along the radial will take place, however. If this desired, it must be specified as the next instruction step.

If an entry is present in the HEAD field, however, the program will assume that upon reaching the specified radial the turn must be started:

PROCEED TO **** **** RAD

One more instruction remains: no entry in ALT, DIST, RAD, or

HEAD but only an entry in NI. This has the following meaning:

PROCEED DIRECT TO NAVAID ****

In that case, the program will find the radial which is tangent to a turn initiated immediately, and having completed the turn move the aircraft to the navaid along this radial. All radial values and headings conform in the Chronicle entries to common aviation practice of giving leading zeros: a heading of 90 degrees will print as 090.

The error messages for headings and radials are similar. No radial or heading should ever be larger than 360 degrees:

ILLEGAL ENTRY TURN ****

ILLEGAL ENTRY **** ****

The navaid identifier must be non-blank for any navaid referenced. The negative value of a radial is only allowed where it makes sense: when intercepting a radial. It is not allowed when a heading is also specified. In that case the the "-" sign will appear in the listing as in:

PROCEED TO NNV - 400 RAD

followed, of course, by an illegal entry error message.

Since an aircraft must first become airborne, before a turn-to-heading instruction is allowed the following error is self-explanatory:

ILLEGAL ENTRY TURN *** AS FIRST INSTRUCTION

In any event, the very first step must move the aircraft off the ground. If this is not the case, the program will print:

```
AIRCRAFT REMAINS ON GROUND DURING FIRST INSTRUCTION
```

after the listing of maneuvers executed.

Having discussed the individual entries on the card, we now check the semantics of combining them. The altitude and distance can appear by themselves on the card with the meaning given earlier. If both appear, the event which occurs <u>FIRST</u> will take precedence. This is evidenced by the noise word "OR" which now appears in the Chronicle:

```
CLIMB TO        **** FT OR
PROCEED FOR   ****** FT
```

where the implication is <u>whichever comes first</u>. If only these two entries are coded, the aircraft will be moved the appropriate distance. If a heading is also specified, however, the aircraft will be turned onto this heading as evidenced by the noise word "then":

```
CLIMB TO          **** FT OR
PROCEED FOR     ****** FT THEN
TURN TO HEADING    ***
```

Matters can be complicated one step further:

```
CLIMB TO          **** FT OR
PROCEED FOR     ****** FT OR
PROCEED TO     ******* RAD THEN
TURN TO HEADING
```

Again the implication is that of whichever comes first. The meaning of the following is not necessarily obvious:

```
CLIMB TO         **** FT OR
PROCEED FOR    ****** FT OR
INTERCEPT ***     *** RAD  (INBOUND)
```

The program will check which event occurs first:

1. Altitude reached
2. Distance flown
3. Turn initiated to intercept the radial

If the first two conditions do not occur before the turn is started to intercept the radial, the third alternative is chosen. If however, either of the first two conditions occurs before the third, this will take precedence and in that case the radial will not be intercepted. In the first two alternatives the aircraft heading will not change, but in the third it will. The resulting flight paths can therefore be very different! The user is advised to obtain a plot of his flight tracks and to check the maneuvers executed listing after each flight listing. If it is desired that the first two conditions are checked first, and to intercept the radial after altitude or distance is reached, the instruction should be split and put on two cards:

```
CLIMB TO         **** FT OR
PROCEED FOR    ****** FT

INTERCEPT ****    *** RAD (INBOUND)
```

Continuation cards, therefore, allow us to construct procedures of more than one step. On occasion there is no difference in the meaning whether one or two cards are used. Since the blank line between two steps has the same meaning as a THEN phrase the following two entries are equivalent:

```
CLIMB TO          ***** FT THEN
TURN TO HEADING   ***

CLIMB TO          ***** FT

TURN TO HEADING   ***
```

In the second case the computer considers the instruction as a two-step instruction, which takes up more space than the one-step instruction of the first method. This "unecessary" step is counted as a step in determining the complexity of the procedure (page 142).

The error messages, which are associated with the instruction part of a DEPART card, but which have not yet been discussed, are generated at the time the procedure is executed for a particular aircraft. Since these messages may be influenced by the presence of restrictions, we will discuss the restrictions first and then come back to the errors.

3. THE RESTRICTION PART OF THE DEPART CARD

The restriction part of a DEPART statement can be used to introduce altitude restrictions. The restriction is of the type "at or below" rather than "at or above." The restriction is interpreted as follows:

```
RESTRICTIONS
UNTIL ***        *** RAD
STAY BELOW      **** FT

RESTRICTIONS
FOR            ***** FT
STAY BELOW      **** FT
```

```
RESTRICTIONS
FOR                 ***** FT OR
UNTIL ***            *** RAD
STAY BELOW           **** FT
```

The NAVAID code for a restriction is placed in the NR (Navaid Restriction) field.

The meaning of the "OR" is different for restrictions than it was for instructions. It signifies that the restriction is in effect until either a radial is crossed or until a distance has been flown, whichever is greater. If an error is detected in the restriction, the message will read:

```
ILLEGAL RESTRICTION -- ALTITUDE =    **** FT
                       DISTANCE =    **** FT  NAVAID = ***   *** RAD
```

The message may appear more than once; it is issued for each item in error. The error pertains to the item immediately preceding the message. A navaid can cause this message when either the NR field is left blank, or the radial is not in the range from 1 to 360. If the navaid name is missing, this will be indicated in the Chronicle message, but if at the same time the radial has an illegal value this will not result in a second diagnostic message.

If an altitude is specified and neither a distance nor a radial is specified it is assumed that the altitude restriction will stay in effect for all distances. This corresponds to climbing to a given altitude and then maintaining level flight. The Chronicle entry reads simply:

```
RESTRICTIONS
STAY BELOW           **** FT
```

and no further message is generated. At the time an aircraft performs this procedure error messages may or may not result depending on the wording of the procedure and the performance characteristics of the aircraft. In any event the altitude profile printed at the end of the procedure will reflect this condition. It is good practice to check each altitude profile generated from a procedure which contained restrictions even when no messages were generated.

The distance in the phrase "FOR **** FT" in a restriction is counted from the position of the aircraft just before the instruction part is executed. Or equivalently, distance references for both instructions and restrictions are computed from the same point: the end point of the previous procedure step.

An altitude restriction, once in effect, will stay in effect until it is satisfied. For example:

```
PROCEED FOR          5000 M

   RESTRICTIONS
   FOR               6000 M
   STAY BELOW         700 M

PROCEED FOR          2000 M
```

will have the following effect. The aircraft will move 5 kilometers at or below 700 meters; the aircraft will then move for another 2 kilometers but during the first 1000 meters the 700 meter restriction remains in effect.

This becomes particularly complicated when a radial is referenced. If the aircraft on its current heading will cross the desired radial, the restriction is clear. It it does not cross the radial, there are two possibilities.

1. The aircraft proceeds away from the radial. In that case the radial reference is deleted. If a distance was also specified, the distance reference will remain. If the aircraft moves toward the radial at a later time, the radial reference will <u>not</u> be reinstated. If no distance reference exists, the entire restriction is deleted:

```
*** *** RAD DELETED FROM RESTRICTION
```

or

```
*** *** RAD DELETED FROM RESTRICTION
    *** FT  RESTRICTION RESCINDED
```

2. The aircraft is not proceeding away from the radial. In that case, the aircraft is either moving parallel to the radial, or the aircraft is on the "opposite" side of the navaid (Figure VI.2). In that case the message is:

```
*** *** RAD NOT INTERSECTED
```

Neither case will generate an ERROR condition, but <u>the user should carefully analyze his Chronicle and satisfy himself that the restrictions are properly incorporated in the execution of the procedure</u>.

We have seen that restrictions are not deleted until satisfied. If the following instruction is read:

```
PROCEED FOR  16000 FT THEN
TURN TO HEADING  260

          RESTRICTIONS
          UNTIL LAX  060  RAD
          STAY BELOW  1500 FT

PROCEED FOR  20000 FT
```

the program may compute that the aircraft will intersect the 060 radial after 60000 ft. No message is therefore generated for the first step. After turning it is, however, possible that the radial is not intersected and the aircraft is not moving away from the radial (in the sense of the program).

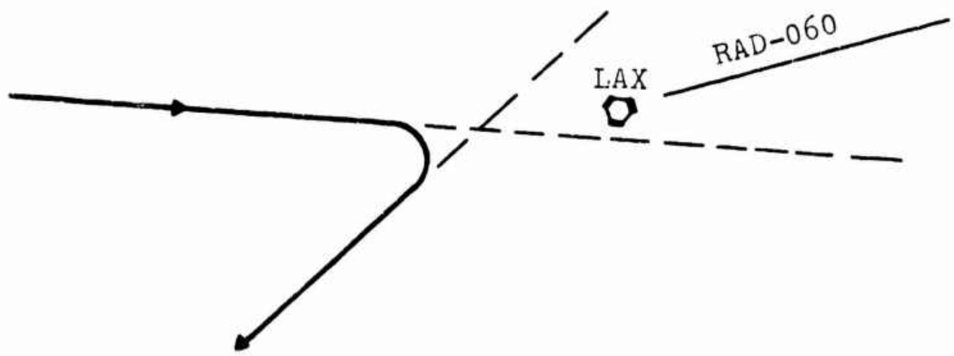


FIGURE VI.2 Aircraft trajectory where radial is not intersected nor does the aircraft move away from radial

In that case, the restriction remains in effect and the aircraft will remain at an altitude not higher than 1500 ft. This may or may not cause an ERROR later (e.g., when an instruction to climb to a higher altitude is given). It is because of such sometimes unexpected interpretations of a procedure that one should carefully check the flight track and altitude profile generated.

Sometimes it is desirable to give a set of altitude restrictions as for instance:

CLIMB TO 3500 T

RESTRICTIONS	
FOR	20000 FT
STAY BELOW	1000 FT
FOR	40000 FT
STAY BELOW	2000 FT
FOR	60000 FT
STAY BELOW	2500 FT

This is communicated to the computer by a DEPART card on which the first restriction appears on the DEPART card itself and the subsequent two restrictions appear on two continuation cards. The instruction field of the continuation card is left blank. A restriction on a continuation card is appended to the last instruction encountered and remains in effect for as many instruction steps as required to satisfy it. If in the above example, the DEPART card had no restriction punched in it, but three continuation cards were used (one for each restriction), the result would have been the same. It is recommended that restrictions are entered for increasing altitudes, but it is not necessary to do so. The three restrictions above could have been entered in any order.

4. FURTHER ERROR MESSAGES

There are many warning and error messages which may be generated in the execution of a procedure. Some of these, referring to navaid references in restrictions, have already been discussed. The following messages are discussed in about the same order as they would be generated by the program.

When a navaid is referred to in a procedure, it must be defined to the program before a FLIGHT card is read. The first

time a navaid is found to be missing, the error message:

NAVAID *** NOT KNOWN

will be printed. On subsequent execution of the departure after a different FLIGHT card, the message will not be repeated unless the first message was:

NAVAID *** NOT KNOWN WILL OVERFLOW IF PRESENT

This message indicates that the navaid was not known at the time; furthermore, the navaid directory is full and no space can be found for the item. This means that a navaid not used in the porcedure must be deleted to make space for the required navaid. A listing will be printed of all navaids known to the program the first time that this condition occurs after space has been previously available in the directory.

The warning:

A/C GONE BEYOND ALTITUDE PROFILE AFTER ** STEPS

indicates that the aircraft has gone beyond the distance for which an altitude profile is defined. The program will keep the aircraft in level flight. This is the same action as that which occurs when the total length of a FLTTRK is more than specified on the altitude profile. In the case of the FLTTRK, no message is generated, however.

If a reference is made in an instruction of an unknown navaid, the program will, when it reaches that instruction, print the basic error message:

NAVAID MISSING ***

This message will repeat for each FLIGHT card where it is appropriate. If an altitude or a distance was also specified, the program will continue laying out the flight track. In that case, the message:

(IT IS IGNORED)

is appended. If this option is not open to the program, the word

ESSENTIAL

appears before the

NAVAID MISSING

message and the procedure is terminated.

A warning is printed when a radial is already past when it appears in the instruction. If no altitude or distance is also specified, the message is an error, and is preceded by the word ESSENTIAL:

(ESSENTIAL) *** *** RAD ALREADY PAST

A similar warning or error can be issued when the radial is not intersected by the current heading of the aircraft:

(ESSENTIAL) *** *** RAD NOT INTERSECTED

The same fixup is taken: if an altitude or distance is present, the radial reference is ignored, else the procedure is terminated.

When an altitude restriction is issued, the aircraft must be at an altitude equal to or lower than the restriction. If this is not the case, the program will issue the following message, after which the procedure is terminated:

ALT. REST. **** FT ILLEGAL AFTER **** FT WHEN A/C IS AT **** FT

This means that either the aircraft performance data do not match the procedure or that the procedure is incorrectly phrased. Errors in coding the cards can also cause this error, of course. Any correction will have to be considered on an individual basis, and no general rules can be given.

A somewhat different variety of the above problem is expressed by:

RESTRICTIVE ALTITUDE *** NOT FOUND IN STEP * (MAX ALT = *** FT).

This message means that the altitude mentioned in a restriction cannot be found in the altitude profile. This may happen, for example, when a navaid reference causes a lower altitude restriction to be extended "to infinity." One should carefully check to see that the procedure is worded correctly and that the restriction chosen by the program is the desired one. The original altitude profile of the aircraft executing the procedure must, of course, extend to include all altitudes referenced in the procedure.

ALTITUDE **** CANNOT BE REACHED

This message may be preceded by the word

ESSENTIAL.

The altitude in an instruction cannot be found in the altitude profile. This may be due to an altitude restriction which is in effect "to infinity" or it may be due to the fact that the aircraft performance data as specified in the altitude profile do not extend to high enough altitudes. The altitude profile should be expanded, the restriction changed or the procedure reworded if the error condition is raised. In the case of a warning, one should carefully check that the procedure was executed as intended.

In that case, the phrase

(IT IS IGNORED)

will appear indicating the decision made by the program. This message is also appended after:

A/C IS ALREADY AT ALT. **** FT WHEN **** FT INSTRUCTION IS GIVEN

This indicates that the procedure does not match the performance characteristics of the aircraft: the climb to altitude instruction comes too late in the procedure. The procedure should be rephrased or it should not be followed by aircraft which climb as fast as those causing the message. If an alternative to climbing to the altitude exists, it will be taken.

When a radial is intercepted, the aircraft may overshoot the radial if the turn radius is too large to make the turn. In that case, the aircraft will ultimately get on course as shown in Figure VI.3.

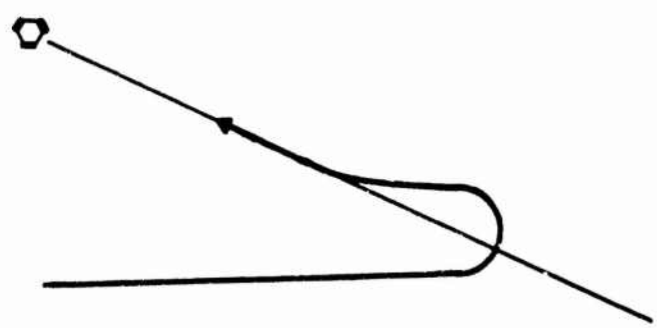


FIGURE VI.3 Radial Overshoot

The message corresponding to this situation is:

A/C TURN RADIUS *** TOO LARGE FOR INTERCEPT OF *** *** RADIAL.
A ** DEGREE COURSE ADJUSTMENT IS MADE

one should carefully check the runway layout plot to see what happened. Correction of the problem, if needed, can take the form of rephrasing the procedure, changing the aircraft performance,

or using a different procedure for the given type of aircraft/ mission combination.

When a proceed direct to a navaid instruction is encountered, the program will locate the navaid with respect to the aircraft. The aircraft will be turned onto the shortest turn to reach the navaid. If the navaid is within the turning radius of the aircraft, the program assumes that the aircraft is already at the desired location and no further action will result. The following message will appear:

```
CANNOT PROCEED DIRECT TO ** NAVAID-DIST = **** FT RADIUS = **** FT
A/C CONTINUES ON HEADING *** DEG
```

one should check that the aircraft has indeed performed the procedure as desired and rephrase the procedure if this is not the case.

When the program finishes generating an altitude profile, this new profile will now become the profile used for this aircraft/ mission combination (for the duration of the procedure only). Although the program can generate a very complex altitude profile, it can only use the first 10 altitude/distance pairs for the subsequent calculation. In other words: the same limitation applies to storing a computer generated profile as applies to a user specified profile. The entire profile will be printed by the program at the concusion of the procedure so that the user can see what happened. If a truncation occurs, it will be signified by a dashed line across the profile listing. The profile will be followed by the message:

```
PROFILE GENERATED EXCEEDS STORAGE AVAILABLE
```

and the program will keep the aircraft in level flight at the altitude specified by the 10th entry in the profile for distances larger than correspond to this entry.

When a procedure is halted abnormally, the following message will appear after the appropriate diagnostic:

PROCEDURE ABANDONED AFTER ** STEP(S)

emphasizing the fact that there remain steps in the procedure which did not get executed and which may well contain further errors.

The only errors which have not been discussed are those associated with the complexity of the procedure. The message printed as a result of a too complex procedure is

PROCEDURE TOO COMPLEX

This message may or may not be appended with an identification of a specific type of data. Table VI.1 associates the allowable complexity with the messages printed.

It should be mentioned that the number of distances, altitudes and navaids means the total number of references for instructions and restrictions combined. If a restriction and an instruction reference the same distance, for example, each will still generate a separate reference. Two references are generated in this case since the "compiler" of the procedure will not check the "symbol table."

TABLE VI.1

ALLOWABLE COMPLEXITY OF PROCEDURES

ITEM	TOTAL NUMBER ALLOWED	MESSAGE
STEPS	15	PROCEDURE TOO COMPLEX (immediately before an instruction)
RESTRICTIONS	15	PROCEDURE TOO COMPLEX (immediately after the word RESTRICTIONS)
ALTITUDES	25	PROCEDURE TOO COMPLEX ALTITUDES
DISTANCES	25	PROCEDURE TOO COMPLEX DISTANCES
NAVAIDS	15	PROCEDURE TOO COMPLEX NAVAIDS
HEADINGS	15	PROCEDURE TOO COMPLEX HEADINGS

It should be mentioned that in general, SIDs, when translated into flight track lengths, are much larger than the area of interest for NEF contours. The user can save considerable computer time by not specifying those parts of a SID which are clearly outside the range of interest. For example, consider

```
CLIMB TO  2500 FT THEN
TURN TO HEADING 160

INTERCEPT NNV 050 RAD (INBOUND)
INTERCEPT MMV 158 RAD
```

If this instruction causes the aircraft to move far away, as shown in Figure VI.4, a great amount of unnecessary calculation is performed. It would be more advantageous to specify

```
PROCEED FOR 60000 FT
```

instead of the two (strictly speaking) unnecessary further instructions.

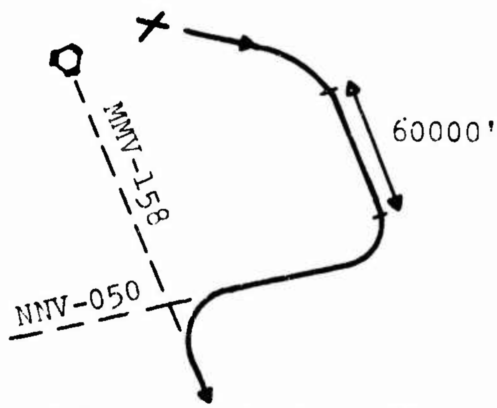


FIGURE VI.4 Example of Excessively Long Flighttrack in a Procedure

This saves time in that less geometric work has to be done, but also a considerable saving in computer time is realized in computing the grid. A factor of 2 to 5 gain in execution speed may be obtained when the procedure is limited to only those steps which are in the immediate vicinity of the airfield. The immediate vicinity is considered to be a radius of approximately 8 miles around the airport. As an aid in assessing after a checkout run how much of a procedure can be deleted, a listing of total distance covered is provided following each FLIGHT entry in the Chronicle.

The DEPART statement constitutes a very powerful method of entering flight tracks into the computer. It is, however, at the same time capable of creating much more data than is necessary or desirable. Only experience on the part of the user will show him what amount of detail forms the compromise between minimizing computer time and retaining all necessary features.

5. USING DEPART FOR VFR OPERATIONS

There is one item which should be pointed out. Although the DEPART statement is worded in such a way that an instrument departure is implied (IFR), the same method can be used to enter VFR instructions. To do this one introduces what might be called "pseudo-navaids."

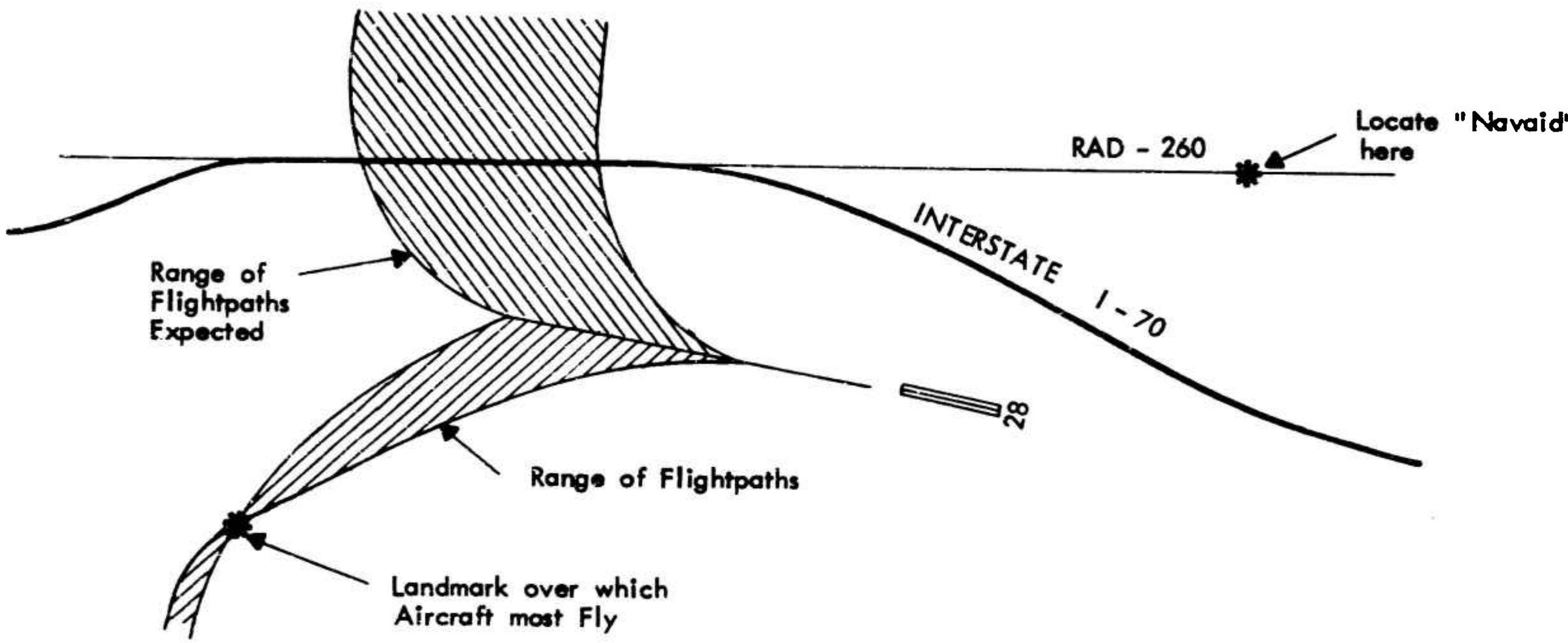
When a pilot's instruction is "climb to 1500 ft. then proceed to the power plant," this is essentially the same as climbing to an altitude and then proceeding direct to a navaid. In the one case the navaid location is determined by looking on an instrument, in the other by looking out of the window. The resulting flight path is not different and one may well use a factory or other landmark as a "navaid" for a VFR clearance.

Radial references may also be used. For instance, a "navaid" and "radial" may be selected to closely correspond to the location of a major highway, etc. A clearance containing a directive "cross the interstate at or below 2500 ft" can then become a restriction:

```
UNTIL I-70     260 RAD
STAY BELOW    2500 FT
```

where the 260 radial of a fictitious navaid I-70 corresponds to the section of freeway where the aircraft will cross (c.f. Figure VI.5)

The user can, with the above suggestions, readily make up a large variety of VFR "procedures." Since these procedures are "invented" by the user, they must be thoroughly checked. For published SIDs at least the user has a fair amount of confidence that the procedure makes sense for aircraft of certain performance characteristics. When a pseudo-procedure is used, the user must check the appropriateness of the instruction for the aircraft as



2 POSSIBLE USES OF A DEPART STATEMENT FOR VFR OPERATIONS :

FIGURE VI-5. USING DEPART FOR VFR CLEARANCES

well as the validity of the instructions. Nonetheless, the user is encouraged to gain experience in formulating VFR procedures in terms of the DEPART statement, since it is a valuable tool in modeling airfields. The only limitations are the allowable instruction set and the imagination of the user.

6. THE GENERATION OF AN ALTITUDE PROFILE

When a procedure is specified, the program will generate a new altitude profile for each aircraft/mission combination as it is required. The new profile may be identical to the original profile, but it may be very different.

The reading of a FLIGHT card will initiate a search for the proper altitude profile. If this flight is taking place on a flight path specified by a procedure, the program will generate a new altitude profile. The new profile is generated "along with the aircraft movements." That part of the profile which corresponds to distances from start of takeoff roll less than or equal to the distance flown by the aircraft is frozen. The profile beyond the current aircraft position can be altered.

The new profile starts off with one point specified: at distance zero the aircraft is on the ground. The following steps are repeated for as many instructions as are in the procedure. (See Figure VI.6.)

1. The current altitude is located in the original profile and the corresponding distance is found.
2. The current distance from start of takeoff roll is located in the new profile.
3. The new profile is cut at the point where the aircraft is at this moment. Distances larger than the current position are dropped.

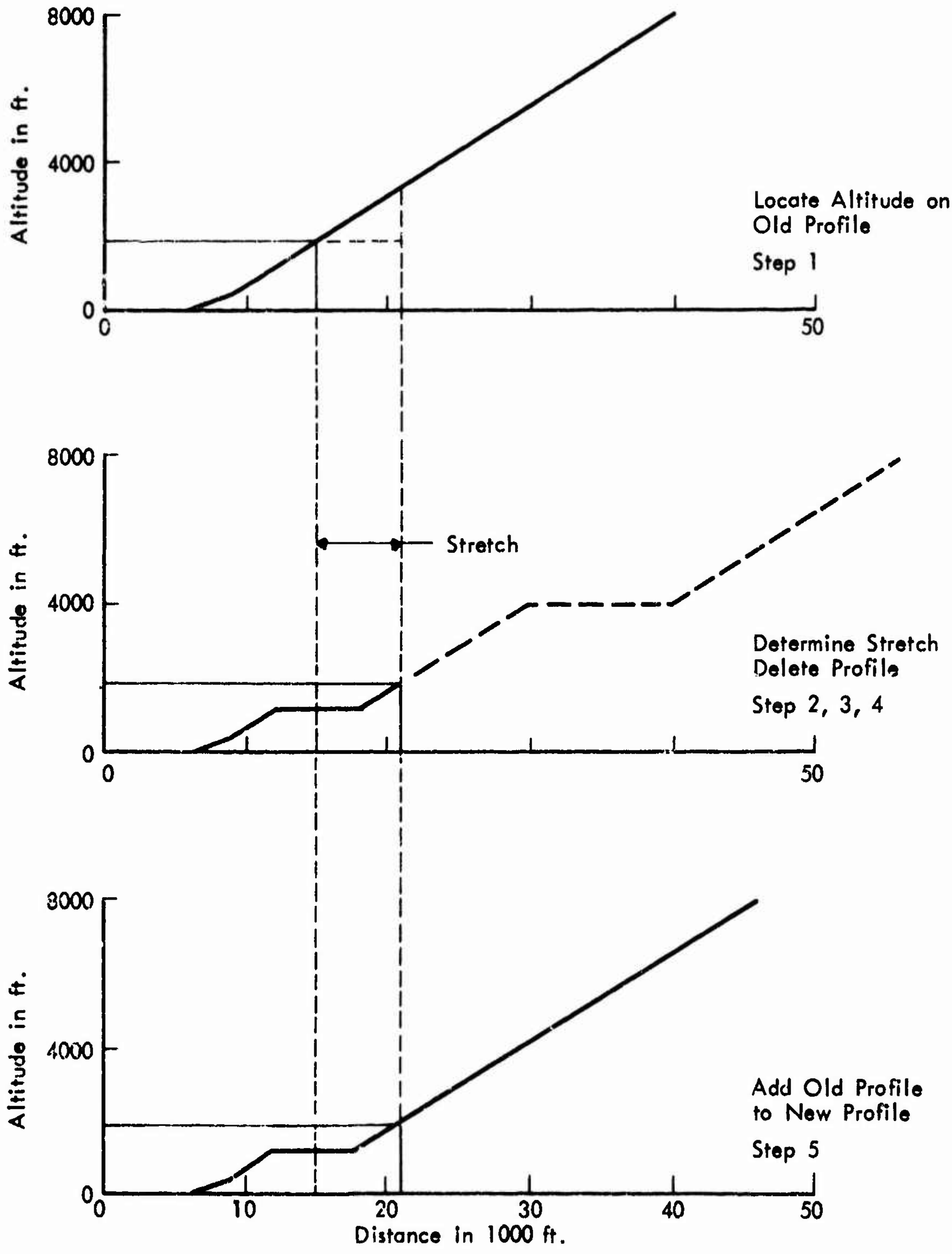


FIGURE VI-6. GENERATION OF ALTITUDE PROFILE

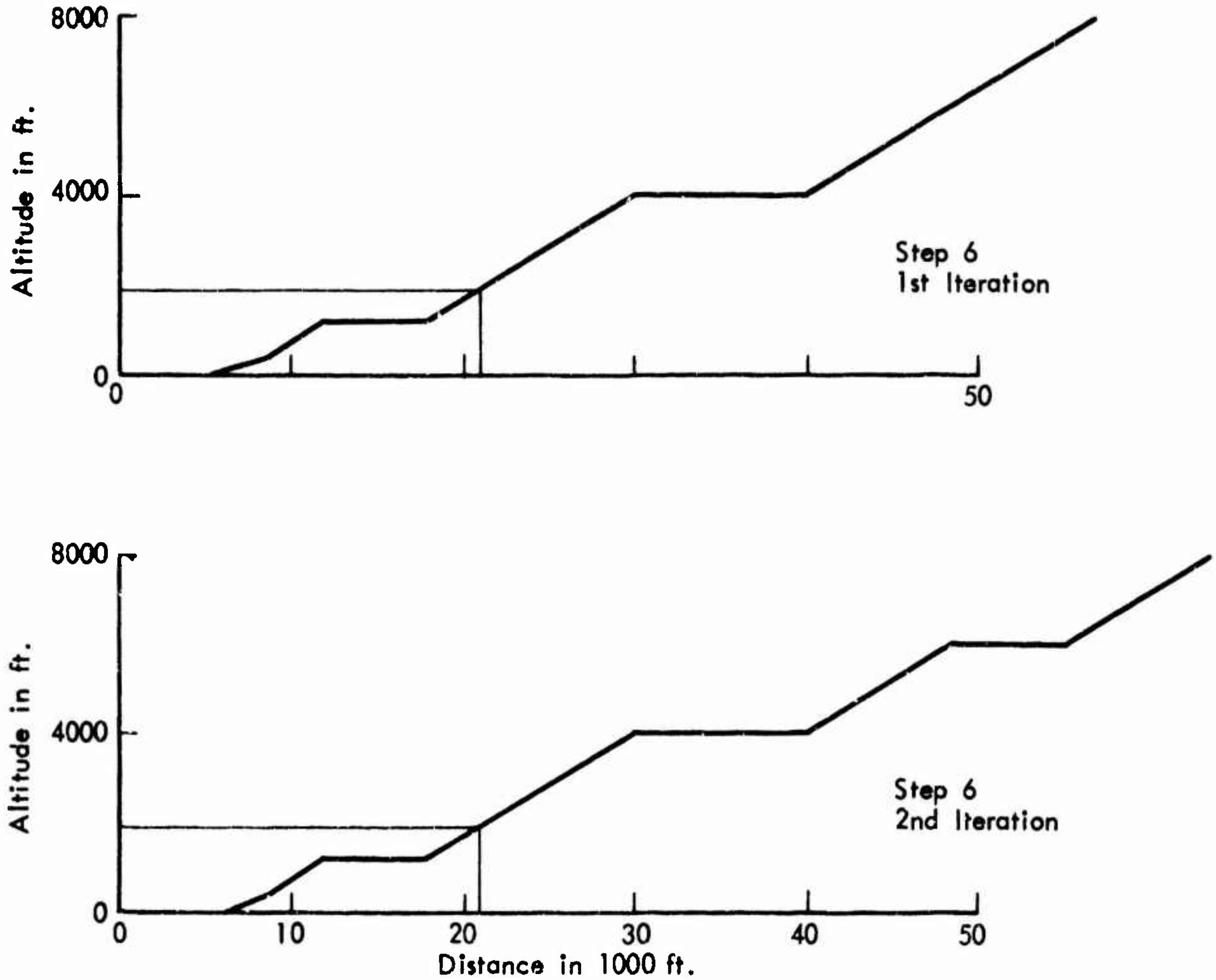


FIGURE VI-6. CONTINUED

4. The "stretch" of the old profile is computed. The stretch is the difference between the distance actually flown minus the distance which the aircraft would have flown to get to the same altitude using the original altitude profile.
5. The part of the original altitude profile which contains altitudes larger than the current aircraft altitude is spliced behind the truncated new profile. (The proper amount of stretch is introduced along the distance axis.)
6. Altitude restrictions which are currently in effect are located. Each altitude is located in the new profile. The amount of stretch which must now be used to expand the new profile is computed. If the stretch is zero or negative, the altitude profile will automatically satisfy the restriction. If the stretch is positive, the aircraft after reaching the altitude, will be kept level at that altitude for the distance given by the stretch. All points further along on the profile are extended by the same amount.

The reason that the altitude restrictions for points beyond the current aircraft position are recomputed every time is related to the fact that radial references may occur. If one occurs, the distance along the flight track where a radial is intersected depends on the aircraft heading. If during an instruction the aircraft heading changes, the distance to the radial may vary. The altitude restriction therefore may manifest itself in a different way each time the aircraft heading changes.

LEWIS THREE DEPARTURE FALCON AFB

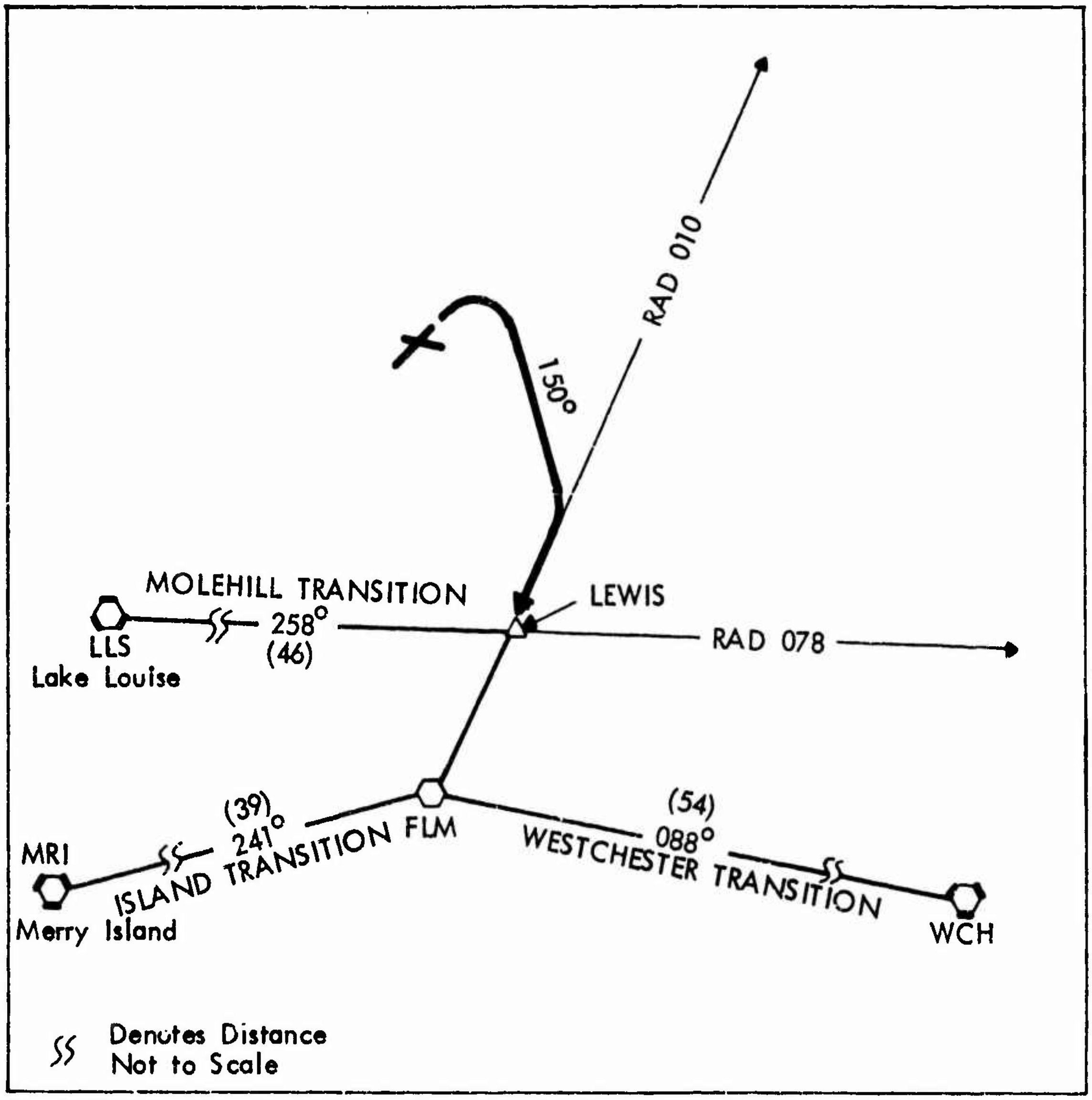


FIGURE VI-7. LEWIS THREE DEPARTURE

7. THE LEWIS THREE DEPARTURE

The Lewis Three Departure (Figure VI.7) can be used as an example of aircraft operations. The aircraft must turn 1 NM beyond the runway. Since the runway itself is 11000 ft., this distance is 17000 ft. The aircraft then turns to a heading of 150 and intercepts the FLM 010 radial. There are then three options:

1. Molehill Transition:
   Intercept and proceed inbound along the LLS-078 radial.
2. Island Transition:
   Intercept and proceed inbound along the MRI-061 radial.
3. Westchester Transition:
   Proceed outbound along the FLM-088 radial.

When one locates FLM on the map, we see that it is well below the lower edge of the square of interest. Therefore the Island and Westchester transitions can be ignored since they do not contribute to the NEF values but would take considerable computer time.

The Molehill transition takes place over the grid of interest and should therefore be considered. Since the LLS VORTAC is 46 miles away, there is no need to consider all 46 miles of flight track. It would be more appropriate to limit the flight track to about 10 miles.

We see that we are therefore left with two procedures. The first defines the flight track for the Molehill transition, the other reflects the other two transitions combined up to FLM. Since the instructions are different, one must consider them as two <u>independent</u> departure procedures. The Molehill transition becomes:

```
PROCEED FOR                 17000 FT THEN
TURN TO HEADING               150
INTERCEPT FLM                 010 RAD   (INBOUND)
INTERCEPT LLS                 078 RAD   (INBOUND)
PROCEED FOR                 60000 FT
```

The other two are:

```
PROCEED FOR                 17000 FT THEN
TURN TO HEADING               150
INTERCEPT FLM                 010 RAD   (INBOUND)
PROCEED DIRECT TO NAVAID FLM
```

It is fairly clear that these procedures will generate a considerably longer flight track than is strictly necessary. This is one of the things which will show up in a NOPROC run when actual distances can be found. It may be desirable to review the NOPROC listing and delete procedure steps after checking the listing of the flight track for all aircraft following the procedure.

SECTION VII

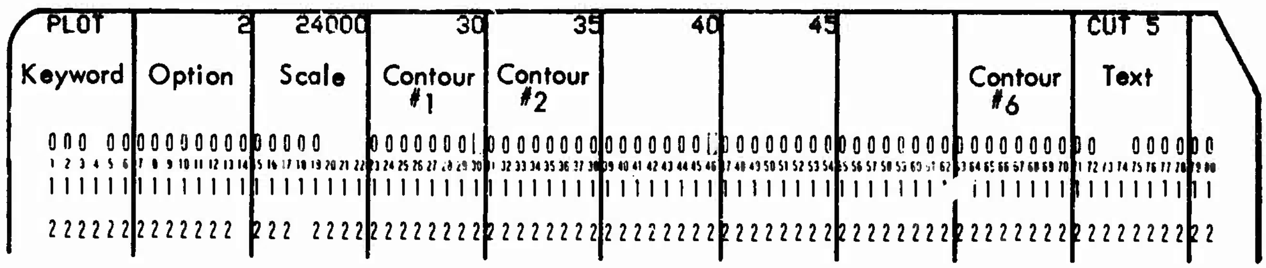
NOISE EXPOSURE FORECAST CONTOURS

1. THE INTERFACE WITH GPCP

The ultimate output desired from NEFUSAF will quite frequently be a set of NEF contours. It is possible to do the contouring by hand. If NEFUSAF printed the grid values (DMPGRD with the PRNT option, or PRINT card), this listing may be spliced together to yield a map of the grid values through whcih contours may be drawn by hand. On page 11 it is shown how one may put such a map together. Although the necessary calculations may be performed by hand, it is generally more efficient to have a computer program claculate the contours and plot them on a graphic output device.

NEFUSAF can produce output which is compatible with GPCP, a General Purpose Contouring Program, available from California Computer Products (CALCOMP). We will not discuss the algorithms used by GPCP, nor do we intend to indicate how storage requirements for GPCP can be made of the right size to fit the NEFUSAF output. These matters are fully discussed in the CALCOMP manual. The CALCOMP manual also describes additional features of GPCP which may be useful in certain contouring situations. Such features, although useful at times, are outside the scope of this volume.

The writing of a GPCP compatible file is initiated by the NEFUSAF control card:




```
+++  PLOT   CUT 5   WIDTH ** IN, OPT = 2, SCALE 1 TO  24000 (1 IN =  2000 FT
                    GPCP CONTROL CARDS ON UNIT 11
                    CONTOURS ARE  --   30   35   40   45
                    THERE ARE  *** ANNOTATION RECORDS  **** NEF DATA POINTS
```

The input to GPCP can be regarded as two sets of data cards. The first set contains control cards which allow the user to specify certain operations to be performed. These cards indicate what contours are desired, the scale of the plot and whether to draw a border or to stop the plotter for pen changes, etc. The second set of cards contains the values of the individual (random) control points to be contoured. Since NEFUSAF may produce in excess of 10000 data cards for each "PLOT" card, it is more efficient to write this information on a magnetic tape.

Although the control card part of the GPCP compatible output can be made available to the NEFUSAF user on punched cards, it is for most purposes unnecessary to punch them and they can be included on the tape file. In that case, only 3 cards will be punched. It is recommended that the card images be written on tape; in this way there is never any problem to replot contours since the data and the control cards are together on the tape and cannot be separated.

2. PLOT OPTIONS AND DEFAULT PARAMETERS

The user has several options at his disposal, even within the confines of a "standard" NEFUSAF/GPCP run. These options are specified in the OPTION field of the PLOT card and listed in Table VII.1.

TABLE VII.1

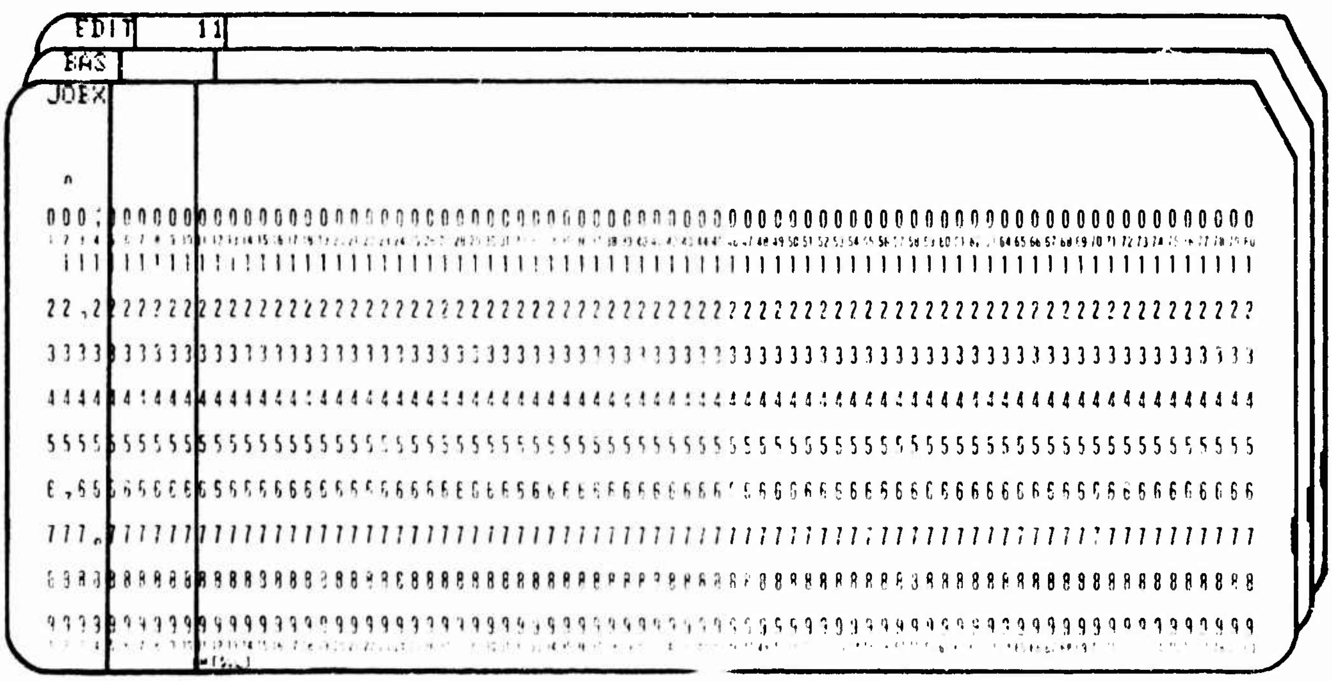
PLOT OPTIONS

OPTIONS	ACTION TAKEN BY THE PROGRAM
± 1	Mark grid points on the map
± 2	Do not mark grid points
± 3	Mark grid points and post local NEF value
± 4	Same as (1) but suppress flighttrack map
± 5	Same as (2) but suppress flighttrack map
± 6	Same as (3) but suppress flighttrack map

If the option is positive, all GPCP control cards other than the first three will be put on the same tape as the NEF data cards. If the option is negative, the GPCP control cards will be punched on the card punch (unit 8).

The tape on which the GPCP compatible output is to be written is always associated with logical unit 11. The user who wishes to avail himself of the more advanced features of GPCP can edit the punched cards to suit his particular needs.

When a tape is generated during a run in which positive and negative options were intermixed, it is important to remember that the tape can only be processed by GPCP with the punched control cards of that particular run. In general, each tape on which files have been written which have punched decks associated with them should never be separated from such decks, since only the combination of the tape with its complete associated deck will give a correct GPCP run. Any tape, which contains only data written with a positive plot option, can be plotted using the following 3 cards:



These three cards are the same as those punched out by the NEFUSAF program during a normal run. Except that the the JOBX punched by the program will have the name of the first airfield punched in it for easy identification of the corresponding tape (if tape and cards are kept together).

If the user leaves the OPTION field blank, the program will default to option = 2. This will not cause any error or warning message, in keeping with the basic philosophy of the program that all standard defaults will not result in diagnostics. If any option other than the set form -6 to +6 is used, the program will issue a warning and set the option to the value 2.

When an option with absolute value 1, 2 or 3 is selected, the program will produce two sets of maps, which can be plotted on top of each other or which may be plotted on separate sheets to make overlays.* The first plot produced is a set of NEF contours as specified below. The plotter will then draw a border and stop. At this point the operator may start on a fresh sheet of plotter paper or he may continue on the same sheet; when continuing on the same sheet he may wish to change pen weight or ink color. When the plotter is restarted the program will draw a layout of all runways, flighttracks and runup areas used by the program.

These "annotation records" are put on as follows: All runways will appear on the map with a width equivalent to 250 feet. If a runway number was given on the RUNWAY card, it will appear on the map. A displaced landing threshold will be shown as a bar across the runway at the point where the threshold is located. All flighttracks originating on this runway will be drawn. This includes the flighttracks explicitly defined on a FLTTRK card as well as those generated by the computer from a procedure. This process is repeated for all runway and flighttracks.

---

*Online plotters may not be capable of stopping to change paper.

If ground runup areas were entered on a RNPPAD card, the location will appear on the map as an aircraft outline. The orientation of the outline is the aircraft orientation assumed on the pad. The exact location of the pad as specified on the RNPPAD card is given by the apex of the V of the tail. Any NAVAID which was entered will appear on the map as a star (*) with the code underneath. Any of the annotation records described above will not appear if it falls outside of the map area. The entire flighttrack map is omitted if the user specifies an option with absolute value 4, 5, or 6.

The plots can be made to any arbitrary scale smaller than 1:5000. If it is attempted to make a plot to a scale larger than 1:5000, the program will issue a warning and set the scale to 1:24000 (corresponding to 1 inch equals 2000 feet). The desired scale is communicated through the SCALE parameter on the PLOT card. The scale parameter contains the denominator part of the scale. If a scale of 1:36000 is desired, the SCALE field on the card should read 36000. If the scale field is left blank, the scale defaults to 1:24000 without the issuance of a WARNING.

The annotation records which we have discussed are sensitive to the scale of the plot. Not only does the flighttrack layout scale properly, but the labeling of the runways, etc., will also scale up and down, except that no character will be plotted which is smaller than 0.07". The reason for this is simply the limitation of the plotter hardware; when the characters become smaller than 0.07" they become illegible due to the limitations of the plotter resolution and due to the finite width of the pen.

The desired contour values can be inserted on the fields designated "contour values." The first contour value field is unique: if this field is left blank the contours from 25 through 50 NEF will be plotted in increments of 5 NEF. The PLOT data card can be continued on as many continuation cards as are required to specify all desired contours. The use of a continuation

card is fundamentally different from the use of a new PLOT card. In the former case more contours are plotted on the same map, in the latter a new plot is started.

The options with absolute value other than 2 or 5 merit careful consideration. If it is desired to indicate where exactly the grid points are located on the map, option 1 or 4 can be used. In that case the plotter will mark each grid point by a plus symbol (+). When the posting of the NEF value of each grid point is desired, this may be accomplished by the use of an absolute value of 3 or 6. It is recommended that this be done only if there is an urgent need to do so since the posting of NEF values on the plotter is excessively time consuming.*

3. THE USE OF A PLOT CARD IN THE 'NOPROC' MODE: CHECKPLOTS

The data input to NEFUSAF is very complex and in order to assist the user in carefully checking his data, the 'noproc' mode of the program is provided. Although all processing is suppressed during a 'noproc' run, it is still very useful to allow the PLOT control card to write a plot. The plot of the airport layout and flighttracks will still be made even though the contouring part of the GPCP run will be suppressed and no grid information is communicated. This enables the user to make a "checkplot" of his input allowing him to overlay the plotted flighttrack map over the original base map to verify that all geometric data were correctly added.

---

*In the unlikely event that a map of NEF values is required without plotting any contours, a very high NEF value should be specified on the first contour location (>100). Maps of this sort are more efficiently generated on the printer.

All messages will appear in the Chronicle during a 'noproc' run exactly as they would during a 'proces' run. A very much smaller file will be written to logical unit 11 if the standard options are specified. If a negative valued option is given, there will be no output on unit 11 since the grid will not be written to unit 11 during NOPROC node. The checkplot provides the user with one of the most powerful tools to check his flight-track and runway data.

If the user does not want to have the runway layouts appear on his final map, he will have specified an option number with absolute value larger than 3. If a checkplot is made this would appear to result in a plot with no data on it whatsoever: the option suppresses the geometric data, and the 'noproc' mode suppressed the NEF values. If the program is in the 'noproc' mode however, the runway layout will always be plotted. This allows the user to use the same PLOT control card for the checking in the 'noproc' mode as he will use later to generate the desired contour map.

If the program is forced into the 'nogo' mode due to an error, the presence of a PLOT control card will have the same effect as during a 'noproc' mode run; only the runway/flighttrack map will be plotted, irrespective of the option selected. This is consistent with the concept of giving the user the maximum amount of diagnostic information during each run. This has the consequence that when such a run is restarted, a new volume must be used on unit 11.

4. PUTTING THE MAP TOGETHER

The output from GPCP is a graphic display. If this is written on a plotter GPCP will produce a map which may consist of more than one sheet of plotter paper. The size of each section of the plot is determined by the width of the plotter drum.

If no contrary instructions are given NEFUSAF will assume a paper width of 28 inches. This allows some space for annotation and splicing for a 30 inch drum. The size of the plotter page can be set by the user to any other value by means of the WIDTH card. The width should be in available inches of plotter paper. It should be pointed out that if a small drum plotter is used an inordinate amount of splicing must be done to arrive at a complete map. Additionally it becomes very difficult to maintain adequate registration of successive sheets since some stretch will invariably occur in the paper even if the humidity is tightly controlled.

If plots are made on a flatbed plotter, the user should satisfy himself that the plotter will be able to handle the size of the map without hitting any limit switches. This means in practice that if the flatbed plottable width is W inches, the largest scale allowable is the one for which:

$$SCALE = (1.2 * 10^7) / W$$

This is due to the fact that GPCP is designed for use with drum plotters and therefore the X-direction is assumed to be infinite. The program will plot the total X-direction and as much as can be accommodated of the Y-direction on the given paper width. The next sheet will contain the next higher band of Y-values. This procedure is totally unworkable on a flatbed where the paper cannot move under program control.

Many additional features are available from GPCP. The interested user is referred to the CALCOMP manual for these. The NEFUSAF output is complete, however, in the sense that a user need not know anything at all about the control card structure of GPCP as long as he runs all contouring jobs with the data provided by NEFUSAF.

5. DIAGNOSTIC MESSAGES

During the execution of the program, the GPCP control cards may be punched on the card punch (unit 8) or they may be written on unit 11 which also contains the grid data cards.* The program will print in the Chronicle the unit on which the control cards are written. If the unit on which they appear changes, this will also be indicated. The deck, which is punched on unit 8, will contain all the parameters necessary to switch the input routines of GPCP back and forth between the two files and no further user action is required to ensure that a deck and its corresponding tape can be processed by GPCP. The messages are:

GPCP CONTROL CARDS ON UNIT **

GPCP CONTROL CARDS ON UNIT ** TRANSFERRED FROM UNIT **

Since the GPCP routines will always start from a card input, a run which is entirely written to unit 11 will always display as the first message:

GPCP CONTROL CARDS ON UNIT 11 TRANSFERRED FROM UNIT 8

since the three cards necessary for the GPCP initialization will always be punched.

The PLOT card cannot give rise to an error, but several different warnings may be issued. These are associated with missing continuation cards, illegal plot options, scale too large or request for contours lower than 25. The messages dealing with contour values lower than 25 NEF require further clarification. The problem centers around the interpretation and calculation of such contours. The meaning of NEF contours lower than 25 in terms of land planning is not very well defined. If one requests a contour lower the 25 NEF, the program will issue the warning:

CONTOURS BELOW 25 ARE NOT CONSIDERED RELIABLE

*Plot options are given in Table VII.1.

A related problem is that contours below 15 are not valid since the program will not compute grid values less than 7 NEF. When various flights are accumulated on a grid, it quite possible that contour values lower than 15 NEF will result in incorrect contours. Since they are of doubtful value even when computed correctly, these contours will be suppressed. Since a value lower than 15 NEF is also lower than 25 NEF, the message will read:

CONTOURS BELOW 25 ARE NOT CONSIDERED RELIABLE
FURTHERMORE CONTOURS BELOW 15 ARE SUPPRESSED

If all contours desired are suppressed, the program will issue the warning:

ALL CONTOURS REQUESTED WERE SUPPRESSED

The program will print in the Chronicle the number of control points and annotation records:

THERE ARE **** ANNOTATION RECORDS ***** NEF DATA POINTS

In general, this is a for-the-record message. It may, however, appear as a warning when fewer than 100 control points would be written to tape 11 or if no annotation records are present. When zero annotation records and no control points are found, the program has found an empty grid. This would occur if a PLOT card follows a CLRGRD card. The annotation records can also be missing when a CLRGRD card is followed only by FLIGHT cards before a PLOT card is encountered.

The fact that at least 100 control points are needed is due to the limitations of GPCP. Plots with fewer than 100 points are ignored as if the program was in 'noproc' mode: only the flighttrack map is plotted. If the number of data points is very low, it is quite possible that nonsense contours will still appear around the edge of the map.

When options of absolute value > 3 are specified the message will read:

RUNWAY LAYOUT SUPPRESSED **** NEF DATA POINTS

The remaining, self-explanatory messages are:

PLOT OPTION ******* IS ILLEGAL, A VALUE OF 2 IS ASSUMED

SCALE 1 TO ******* ILLEGAL, SET TO 1 TO 24000

CONTINUATION CARD MISSING

6. SCALLOPED CONTOURS

The NEF is a logarithmic unit and therefore its lowest value is minus infinity. The lowest value communicated to GPCP is 10 NEF. The NEF concept also includes a cutoff at 75 EPNdB. If an aircraft causes an exposure on the ground of less than 75 EPNdB, the contribution is ignored. When the volume of operations is very high, it is possible that the 75 EPNdB cutoff corresponds to a sizeable NEF value. When this cutoff occurs at an NEF value for which contours are desired, the contouring program finds a large gradient between the lowest grid value calculated and the empty grid values. The result is that the program draws a wildly scalloped contour along this gradient. It is partly for this reason that contours below 15 NEF are suppressed. The problem should not occur for a 25 NEF contour unless exceptionally high numbers of aircraft are involved (2000 operations per day per runway of which 500 are at night).

Another "problem" occurs close to the runway. For many cases one will find that the higher NEF values (50 or 55 NEF) will show islands over the runway. This is due to the fact that the NEF value close to the (extended) runway centerline may change quite significantly due to transitions from ground-to-ground to air-to-ground propagation. When operations take place in both directions, as many as 5 "islands" may appear along

the runway. Although such islands may be correct in certain cases, one must remember that the control points are spaced 1000 feet apart, and therefore contours within 500 or 1000 feet of the runway are likely to be inaccurate.

The NEFUSAF program was written for military jet operations and the control point spacing is based upon that consideration. If airbases with a very few, very quiet aircraft are considered, the contours of interest in land planning (20 - 40 NEF) may be too close to the runway for adequate resolution as explained above. At that point one has exceeded the design limit of the program and one must change the program grid spacing in order to achieve a higher resolution.

SECTION VIII

RESTARTING NEFUSAF

1. CAUSES FOR RESTARTS

Despite the extensive error checking provided in the NEFUSAF program, it will happen nonetheless from time to time, that the occurrence of an error necessitates a rerun of at least part of the data. If binary dumps are available, one may restart a run from such a dump and conserve computer time, since grid values which are not affected by the error can be copied from the binary dump. It is also possible that errors occur which cause termination of the job by the computer operating system. These errors may be due to keypunch errors, causing an abnormal termination of the job due to conversion errors, or may be entirely dependent on the computer installation. Errors of the latter type are tape parity errors and hardware malfunctions, data set allocation problems, exceeding time limit, etc. At any time that an NEFUSAF error occurs during the 'proces' mode, the program will provide a dump on logical unit 10 which can be used for restarts. Operating system errors do, of course, not provide this courtesy, and one must resort to a user generated dump. Since every error has its own unique recovery, it is impossible to give examples for every possible contingency, but it is possible to give some broad guidelines as to how one might proceed.

The user must first recognize that there are several classes of errors, even within the scope of NEFUSAF errors. There are clearly errors which are flagged by the program with an ERROR message in the Chronicle file. At that point a dump to unit 10 is initiated (provided the program is in the 'proces' mode), and one can in principle restart from this dump after the error is corrected. There are some reservations implied in the above statement however. It is clear that an error may have occurred at a much earlier point than where it was recognized as such by the program. If the error has caused the grid

of NEF values to be updated with incorrect information, the dump initiated by the ERROR routines is useless since it, also, contains erroneous information. In that case, one must find the last dump which was unaffected by the presence of the error.

It is equally possible that the error occurred somewhere in the data, but was never detected by the program as an error. This might occur when a takeoff or landing descriptor card was used which contained the wrong references for altitude, delta-EPNL, or EPNL profiles. If this is the case and the referenced profiles are properly defined the program will never detect this as an error, even though clearly all calculations involving this aircraft/mission combination are wrong.

Finally there are errors which are due to files having been diverted to unit 10 rather than being written on the units intended because of dump count errors or mixing of formatted and unformatted dumps in a previous AIRFLD. Such dumps on unit 10 cannot be read back from this unit. Such runs may require a separate tape reformatting run before they can be restarted for complete processing.

2. THE PROCES/LODGRD COMBINATION

In Section II we have seen that the program maintains several file status indicators which contain information on whether the unit contains formatted or binary information, whether a binary file has been written on or only read, and the position of the read/write heads in a binary file. If one makes a write reference to a file during the 'noproc' mode, no information is transferred to the external medium, but only a file counter is updated. This particular counter is the "logical" counter: the counter which keeps track of how many references are made to the file. The "physical" counter, which is the counter of actual writes

performed, will not be incremented. If one goes from 'noproc' to 'proces' mode and attempts to write on a file where the logical and physical counters are not equal, the program will issue a warning message:

```
UNIT ** CONTAINS *** LOGICAL BUT ONLY *** PHYSICAL DUMPS
                                         SAVED DUMP *** ON UNIT 10
```

as discussed in Section II. When rerunning a job to correct errors, one obviously wants to rerun only those parts of the data deck which need to be recomputed. The ideal way to do this would be to run in 'noproc' mode until some computations must be performed then switch to 'proces' mode to process the corrections. If one does this, it is clear that any external files which may be referenced have physical and logical counts which are different, and therefore this would cause a new error. The PROCES/LODGRD combination is designed to meet this particular need.

It is here that the phrase "restart from a dump" becomes meaningful. One may run the to-be-restarted job in 'noproc' mode up till the last "good" dump of the run which failed. Between the DMPGRD card which generated the dump from which one restarts and the first FLIGHT card one places the pair PROCES/LODGRD with the LODGRD pointing to the file from which the restart is made. This combination of cards has a greater meaning than just the two functions called forth by the keywords. It is also recognized as a RESTART condition, and as such resets all physical counters for external files equal to their logical values. If the same reels of tape are therefore used as were written on during the original run the situation during the restart appears to the program exactly as it did during the original run.

The 'proces' mode will now have the effect of overwriting erroneous dumps following the restart and the resulting files and plots will be indistinguishable from those generated if the original run had been flawless. (If the rerun took place on a different day, this will not be completely true since the binary files and plots will have the date coded in them, which is different.) Since the entire file catalog in NEFUSAF is reset to reflect the status of all external files as they were in the original run, it is imperative that the exact same reels of tape are assigned to the same logical units.*

3. SPECIAL CONSIDERATIONS FOR UNIT 10

Unit 10 is a special unit, in that it cannot be read during a NEFUSAF run, furthermore it is a unit on which it may not be advisable to overwrite existing information, since it is possible that good dumps, which one would like to retrieve later on, follow the dump from which to restart. How this can happen can be seen in the following example. Suppose an ERROR occurs causing a dump to unit 10. In the remainder of the airfield there are two references to unit 16. These are ignored since the program is in the 'nogo'mode and therefore only the logical counters are incremented. The next AIRFLD card will clear the error condition and reset 'nogo' to 'proces'. If this airfield also writes on unit 16 this will cause the dump to be diverted to unit 10 since the physical and logical counts no longer match on unit 16. It is therefore unwise to mount the same volume on unit 10 since the restart dump is not accessible and all existing information will be destroyed if unit 10 is written on again! The volume which was unit 10 in the original run must be assigned to a new logical unit and a fresh volume mounted on unit 10 (to take care of eventualities).

*When rerunning a job, one must remember that files which were newly created during the original run are now in existence. On many computer systems this will require some control card change to reflect the file status of "OLD" rather than "NEW."

If the dump on old unit 10 is the only one on the volume, one can take the volume and mount it on a different logical unit, say unit 17. If there is no DMPGRD card in the deck corresponding to this dump, it is clear that this dump was generated by a NEFUSAF subroutine call as if a DMPGRD card immediately preceded the card in which the error occurred. A PROCES/LODGRD pointing to the first dump on unit 17 must therefore be placed immediately in front of the (corrected) card causing the error. Later sections deal with the problems when unit 10 is used as a default binary unit as well as for error routine dumps.

4. RESTART WHEN SUBSEQUENT AIRFIELDS HAVE FILE COUNT ERRORS

It is immediately obvious that the above considerations make it less than desirable to have different airfields write dumps on the same logical unit. It is at times convenient to do so though, because one may wish to limit the number of tapes in use. The following few paragraphs describe the various methods by which the user can extricate himself from the predicaments of diverted dumps on unit 10 during a restart.

Let us for sake of argument again assume that the tape originally mounted on unit 10 is mounted during the rerun on unit 17. The exact unit is not important and any legal unit not referenced during the original run can be used. The first dump on unit 17 is assumed to be the one from which to restart, so in the appropriate place in the data deck one inserts a PROCES/LODGRD pointing to dump 1 on unit 17. (See below how to restart if the restart dump is not the first one.) Since we will restart from this dump all previous PROCES cards must be removed so that the computations up to the restart point are not performed again. The remainder of the airfield is now computed.

The next step is to get the diverted dumps of subsequent airfields back on the units where they belong, if this is desired. Since we are only planning to copy tape files we place a 'noproc' card in front of the first AIRFLD thereby inhibiting processing. In front of each DMPGRD card which resulted in a diverted dump, one now places four cards: a PROCES card, a CLRGRD card, an ADDGRD card pointing to the proper dump number on unit 17, and a DMPGRD pointing to the desired unit. Although LODGRD is logically equivalent to CLRGRD followed by ADDGRD we do not use a LODGRD card since that would constitute a restart which we do not want at this time. The restart condition causes file pointers to be reset, etc., which is not desirable at this point. The DMPGRD card is now followed by a NOPROC card to inhibit further processing until the next diverted dump is to be restored. The process is repeated for all diverted dumps.

The DMPGRD card will produce the desired dump on the desired unit since the logical and physical counters should now be in agreement. Had the restart failed for one reason or another and had we copied the remaining dumps using a LODGRD card, the program would have made the physical counter equal to the logical counter and caused the program to assume that more files were on the tape than were really there. This could cause a NEFUSAF error if the tape contained a recognizable end-of-information record written by NEFUSAF at some previous occasion, or else would result in a system error due to the attempt to read beyond the filemark on the tape.

Because one can never be sure that a previous restart has executed properly all the way to the end it is emphatically recommended that restarts of successive failed runs are not performed during the same rerun, but that they are done one at a time.

It is important to recognize that all preceding cards should be included for restarts of airfields other than the first one, since the file counters must be properly set. An alternate way to achieve this is by leaving all preceding airfields out, but putting in dummy write references to all files which will be written on during the restarted run.

In other words, if the restarted run should require writing dump 6 on unit 14, one puts 5 DMPGRD cards pointing to unit 14 in the 'noproc' part of the deck. These 5 cards take the place of the 5 cards which were present in the preceding airfield(s). The restarted run will then space over 5 dumps before writing the 6th dump.

Since the program resets the counters to bring the physical count up to the logical count, this process will overwrite any dumps beyond the logical count of the tape. If an erroneous volume was assigned to the logical unit it would overwrite any dumps there. As we emphasized before: it is imperative that the same volumes are mounted on the same logical unit so that information is not inadvertently destroyed.

5. MIXED DEFAULT AND ERROR ROUTINE DUMPS ON UNIT 10

If the unit 10 dump from which the restart is made is not the first one, it will generally be true that the preceding dumps were default dumps. One can leave these on unit 17, and not worry about them, or one can copy them to unit 10 so that unit 10 will, at the end of the run, look exactly as originally intended. To do this, one restarts from the correct point but starts at the first dump on unit 17 and copies to unit 10 by means of a default dump using a DMPGRD without a unit number, followed by as many LODGRD/DMPGRD combinations as required to make the file on unit 10 look as intended. This is repeated until the dump from which one desires to restart is read and one can continue

processing. If the dump from which it is restarted was a user generated default dump, the DMPGRD card should be used to copy the dump from unit 17 to unit 10 in order to preserve the original appearance of this tape; if the dump was due to the ERROR routine and not user evoked this last LODGRD should not be followed by a DMPGRD card.

6. RESTART MESSAGES

The PROCES card, when encountered, will cause its normal message to be entered in the Chronicle as usual. The LODGRD immediately following a PROCES card will cause a restart message to be printed indicating which files one considered open to the program:

```
+++  RESTART FROM PREVIOUS ERROR

     EXTERNAL FILES ATTACHED

                    UNIT ** BINARY  WITH *** DUMPS
                    UNIT ** PRINTED WITH *** DUMPS
                    UNIT ** BINARY  WITH *** DUMPS LOGICAL BUT *** PHYSICAL
                    UNIT ** INPUT ONLY
```

The first message appears for each binary unit which is still at the load point. The second message is used for all formatted files. The third message indicates that the number of logical and physical units is not equal, in other words, the tape was written on during the run in which the restart occurred. This is in itself not an error, but it is obviously a dangerous thing to do. The fourth message appears for read-only dumps, i.e., a tape which is being read but has not been written on during the present run. "Input only" files are always rewound and placed at their load point during a restart.

If an illegal header has been encountered on a tape, the restart procedure will give the tape "input only" status:

UNIT ** INPUT ONLY (LIMITED ACCESS DUE TO ILLEGAL HEADER)

It is also possible to get the message:

UNIT ** BLOCKED (LIMITED ACCESS DUE TO ILLEGAL HEADER)

indicating an unexpected end of file on a unit which was written on during an earlier part of the job.

Both these messages indicate that the run has been in the 'proces' mode prior to the restart point. This constitutes therefore, not a restart in the usual sense. Since this kind of operation is not a true restart but apparently an attempt by the user to do something special, we cannot give any further guidelines here.

If an error is detected in the input data before the PROCES/LODGRD is encountered, the restart will fail and the warning message:

RESTART FAILED - NOT ALL ERRORS WERE CORRECTED

will be printed in the Chronicle. The program remains in the 'nogo' mode for the remainder of the airfield.

7. RESTART OF A RUN WHICH CONTAINS PLOT CARDS

If a run contained PLOT cards before the point from which the restart is made, a file will have been written on unit 11. The tape on unit 11 will receive grid information when the program is in the 'proces' mode but only annotation information will

be present during the 'nogo' or 'noproc' mode. Since the presence of a PLOT card in the 'noproc' section of the deck will only cause the annotation information to be transferred, one should mount a fresh volume on unit 11 during a restart. The original tape 11 should be run through GPCP to retrieve any contours up to the point of restart. If the volume is remounted, all contour information up to the point of restart will be irretrievably lost.

<u>A rule to remember</u>: During restarts all user definable files must be remounted as in the original run, all NEFUSAF internal files (unit 10 and 11) should be given a new volume.

REFERENCES

1. Bishop, D. E., *"Community Noise Exposure Resulting From Aircraft Operations: Application Guide for Predictive Procedure,"* AMRL TR-73-105, Aerospace Medical Research Laboratory, Wright-Patterson Air Firce Base, Ohio (October 1973).

2. Galloway, W. J., *"Community Noise Exposure Resulting From Aircraft Operations: Technical Review,"* AMRL TR-73-106, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio (July 1973).

3. Bishop, D. E. and Galloway, W. J., *"Community Noise Exposure Resulting From Aircraft Operations: Acquisition and Analysis of Aircraft Noise and Performance Data,"* AMRL TR-73-107, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio (September 1973).

4. Horonjeff, R. D., Kandukuri, R. R., and Reddingius, N. H., *"Community Noise Exposure Resulting From Aircraft Operations: Computer Program Description,"* AMRL TR-73-109, Aerospace Medical Research Laboratory (September 1973).

5. Speakman, J. D., Powell, R. G., and Cole, J. N., *"Community Noise Exposure Resulting From Aircraft Operations: Acoustic Data on Military Aircraft,"* AMRL TR-73-110, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio (February 1974).

APPENDIX A

COMPUTER OUTPUT EXAMPLE

```
BBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN                                                                                      BBN
BBN               U N I T E D   S T A T E S   A I R   F O R C E                          BBN
BBN                                                                                      BBN
BBN                          NOISE EXPOSURE FORECAST                                     BBN
BBN                                                                                      BBN
BBN                                12/20/73                                              BBN
BBN                                                                                      BBN
BBN         COMPUTER PROGRAM OPERATOR'S MANUAL EXAMPLE                                   BBN
BBN                                                                                      BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBN   UNITED   STATES   AIR   FORCE   NOISE   EXPOSURE   FORECAST   COMPUTER   PROGRAM    BBN
BBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBN
```

```
*** NEW AIRFIELD  NEF TEST CASE FOR NASA LANGLEY COMPUTER CENTER

                  EXTERNAL LOCATION OF GRID ORIGIN  X = 1620000 Y =  619000
                  MAGNETIC DECLINATION   7.0 DEG TO EAST
                  OPTIONS    PROGRAM WILL PROCESS INPUT DATA  (ENGLISH UNITS)
                             DATA BASE CARRIED FORWARD UNCHANGED

                  FILES KNOWN TO PROGRAM
                  UNIT 10 BINARY WITH    0 DUMPS

*** FLIGHT DESCRIPTOR  AIRCRAFT = B-52G         TURN RAD =    6000.0FT
                      OPERATION = TAKOFF        ALT PROF =     511
                       CLASS NO =     51        POW PROF =     511
                     MISSION NO =      1
                      SUBFLIGHT = 1    NOISE PROF =    511
                                   TRACK LIMITS =        0 TO 1000000 FT

*** FLIGHT DESCRIPTOR  AIRCRAFT = B-52G         TURN RAD =    6000.0FT
                      OPERATION = LANDNG        ALT PROF =      51
                       CLASS NO =     51        POW PROF =      61
                     MISSION NO =     10
                      SUBFLIGHT = 1    NOISE PROF =    512
                                   TRACK LIMITS =        0 TO 1000000 FT

*** FLIGHT DESCRIPTOR  AIRCRAFT = KC-135A       TURN RAD =    6000.0FT
                      OPERATION = TAKOFF        ALT PROF =    2512
                       CLASS NO =    251        POW PROF =    2512
                     MISSION NO =      2
                      SUBFLIGHT = 1    NOISE PROF =   2511
                                   TRACK LIMITS =        0 TO 1000000 FT

*** FLIGHT DESCRIPTOR  AIRCRAFT = KC-135A       TURN RAD =    6000.0FT
                      OPERATION = LANDNG        ALT PROF =      51
                       CLASS NO =    251        POW PROF =      62
                     MISSION NO =     10
                      SUBFLIGHT = 1    NOISE PROF =   2512
                                   TRACK LIMITS =        0 TO 1000000 FT
```

```
***  NEW AIRFIELD  NEF TEST CASE FOR NASA LANGLEY COMPUTER CENTER

                   EXTERNAL LOCATION OF GRID ORIGIN  X =  1620000 Y =    619000
                   MAGNETIC DECLINATION   7.0 DEG TO EAST
                   OPTIONS    PROGRAM WILL PROCESS INPUT DATA  (ENGLISH UNITS)
                              DATA BASE CARRIED FORWARD UNCHANGED

                   FILES KNOWN TO PROGRAM
                   UNIT 10 BINARY WITH    0 DUMPS

***  FLIGHT DESCRIPTOR  AIRCRAFT - B-52G          TURN RAD -     6000.0FT
                       OPERATION - TAKOFF         ALT PROF -      511
                        CLASS NO -       51       POW PROF -      511
                      MISSION NO -        1
                      SUBFLIGHT - 1     NOISE PROF -      511
                                      TRACK LIMITS -        0 TO 1000000 FT

***  FLIGHT DESCRIPTOR  AIRCRAFT - B-52G          TURN RAD -     6000.0FT
                       OPERATION - LANDNG         ALT PROF -       51
                        CLASS NO -       51       POW PROF -       61
                      MISSION NO -       10
                      SUBFLIGHT - 1     NOISE PROF -      512
                                      TRACK LIMITS -        0 TO 1000000 FT

***  FLIGHT DESCRIPTOR  AIRCRAFT - KC-135A        TURN RAD -     6000.0FT
                       OPERATION - TAKOFF         ALT PROF -     2512
                        CLASS NO -      251       POW PROF -     2512
                      MISSION NO -        2
                      SUBFLIGHT - 1     NOISE PROF -     2511
                                      TRACK LIMITS -        0 TO 1000000 FT

***  FLIGHT DESCRIPTOR  AIRCRAFT - KC-135A        TURN RAD -     6000.0FT
                       OPERATION - LANDNG         ALT PROF -       51
                        CLASS NO -      251       POW PROF -       62
                      MISSION NO -       10
                      SUBFLIGHT - 1     NOISE PROF -     2512
                                      TRACK LIMITS -        0 TO 1000000 FT
```

*** ALTITUDE PROFILE NAME = 511

TRACK DIST	ALTITUDE
0 FT	0 FT
5000 FT	0 FT
12000 FT	100 FT
48000 FT	5000 FT
200000 FT	20500 FT

*** POWER LEVEL PROFILE NAME = 511

TRACK DIST	REL POWER (DB)
0 FT	4.0
2500 FT	.5
5000 FT	0.0
200000 FT	0.0

*** ALTITUDE PROFILE NAME = 2512

TRACK DIST	ALTITUDE
0 FT	0 FT
6000 FT	0 FT
10000 FT	200 FT
18000 FT	1000 FT
27000 FT	1000 FT
200000 FT	18660 FT

*** POWER LEVEL PROFILE NAME = 2512

TRACK DIST	REL POWER (DB)
0 FT	5.0
3000 FT	1.5
6000 FT	1.0
40000 FT	1.0
41000 FT	0.0
200000 FT	0.0

*** ALTITUDE PROFILE NAME = 2513

TRACK DIST	ALTITUDE
0 FT	0 FT
8000 FT	0 FT
11500 FT	50 FT
23000 FT	1000 FT
31000 FT	1000 FT
200000 FT	17132 FT

*** POWER LEVEL PROFILE NAME = 2513

TRACK DIST	REL POWER (DB)
0 FT	4.0
4000 FT	.5
8000 FT	0.0
200000 FT	0.0

*** ALTITUDE PROFILE NAME = 51

TRACK DIST	ALTITUDE
0 FT	100 FT
36500 FT	1700 FT
60000 FT	1700 FT
90000 FT	4500 FT
180000 FT	13200 FT

*** POWER LEVEL PROFILE NAME = 61

TRACK DIST	REL POWER (DB)
0 FT	-1.0
300000 FT	-1.0

*** POWER LEVEL PROFILE NAME = 62

TRACK DIST	REL POWER (DB)
0 FT	2.9
300000 FT	2.9

*** FLIGHT NOISE LEVEL PROFILE (EPNL) NAME = 511

DIST	EFFECTIVE PERCEIVED NOISE LEVEL GRND-TO-GRND	AIR-TO-GRND
200 FT	133.9	133.7
250 FT	132.5	132.4
315 FT	131.1	131.0
400 FT	129.7	129.6
500 FT	128.2	128.2
630 FT	126.5	126.6
800 FT	124.8	124.9
1000 FT	122.8	123.1
1250 FT	120.5	121.1
1600 FT	118.2	119.0
2000 FT	115.8	116.7
2500 FT	113.2	114.3
3150 FT	110.2	111.8
4000 FT	107.1	109.6
5000 FT	103.5	107.2
6300 FT	99.6	104.6
8000 FT	94.6	101.9
10000 FT	91.0	99.0
12500 FT	86.2	95.9
16000 FT	81.6	92.5
20000 FT	76.3	88.9
25000 FT	70.6	85.4

*** FLIGHT NOISE LEVEL PROFILE (EPNL) NAME = 512

DIST	EFFECTIVE PERCEIVED NOISE LEVEL GRND-TO-GRND	AIR-TO-GRND
200 FT	123.1	123.1
250 FT	121.8	121.8
315 FT	120.5	120.5
400 FT	119.1	119.1
500 FT	117.7	117.7
630 FT	116.1	116.1
800 FT	114.4	114.5
1000 FT	112.5	112.7
1250 FT	110.3	110.8
1600 FT	108.1	108.7
2000 FT	105.5	106.4
2500 FT	102.8	104.0
3150 FT	99.5	101.9
4000 FT	96.1	99.7
5000 FT	92.4	97.4
6300 FT	88.4	94.9
8000 FT	85.0	92.3
10000 FT	81.4	89.5
12500 FT	77.6	86.4
16000 FT	73.4	83.0
20000 FT	68.6	79.8
25000 FT	63.0	76.3

** FLIGHT NOISE LEVEL PROFILE (EPNL) NAME = 2511

DIST	EFFECTIVE PERCEIVED NOISE LEVEL GRND-TO-GRND	AIR-TO-GRND
200 FT	130.9	130.8
250 FT	129.5	129.4
315 FT	128.1	128.0
400 FT	126.7	126.4
500 FT	125.2	124.9
630 FT	123.5	123.2
800 FT	121.8	121.4
1000 FT	119.8	119.5
1250 FT	117.5	117.6
1600 FT	115.2	115.8
2000 FT	112.8	113.9
2500 FT	110.2	111.9
3150 FT	107.2	109.8
4000 FT	104.1	107.6
5000 FT	100.5	105.3
6300 FT	96.6	102.9
8000 FT	91.6	100.3
10000 FT	88.0	97.4
12500 FT	83.2	94.4
16000 FT	78.6	91.6
20000 FT	73.3	88.5
25000 FT	67.6	85.0

*** FLIGHT NOISE LEVEL PROFILE (EPNL) NAME = 2512

DIST	EFFECTIVE PERCEIVED NOISE LEVEL GRND-TO-GRND	AIR-TO-GRND
200 FT	120.3	120.3
250 FT	119.0	119.0
315 FT	117.7	117.7
400 FT	116.3	116.3
500 FT	114.9	114.9
630 FT	113.3	113.3
800 FT	111.6	111.7
1000 FT	109.5	109.6
1250 FT	107.4	107.8
1600 FT	105.2	105.7
2000 FT	102.7	103.6
2500 FT	100.0	101.5
3150 FT	96.6	98.9
4000 FT	93.3	96.7
5000 FT	90.2	95.0
6300 FT	85.5	92.0
8000 FT	82.3	89.4
10000 FT	78.8	86.6
12500 FT	74.9	83.5
16000 FT	70.7	80.2
20000 FT	65.6	76.4
25000 FT	60.5	72.6

```
                           --------------------------
***                          R U N W A Y    14
                           --------------------------
                    LENGTH  11829 FT, GLIDE SLOPE  2.50 DEG, HEADING 145 DEG
                    START( 1638610,  674620)  END( 1644320,  664260)
                    DISPLACEMENTS - TAKEOFF       0, LANDING       0

***  DEPARTURE PROCEDURE

                PROCEED FOR       48000 FT    THEN
                TURN TO HEADING     034
                     RESTRICTIONS
                     FOR          48000 FT
                     STAY BELOW    3000 FT

                PROCEED FOR      200000 FT

                **********  END OF PROCEDURE **********


***  KC-135A        AIRCRAFT NO. =     251     MISSION NO. =        2
                    OPERATIONS - DAY    2.420 , NIGHT     0.000
                       3000 FT RESTRICTION RESCINDED

     PROFILE GENERATED -- DISTANCE   ALTITUDE
                                0          0 FT
                             6000          0 FT
                            10000        200 FT
                            18000       1000 FT
                            27000       1000 FT
                            46592       3000 FT
                            48000       3000 FT
                           201408      18660 FT

     FLIGHT TRACK  --  MANEUVER              DISTANCE       HEADING
                       EXECUTED                AFTER COMPLETION
                    PROCEED       48000 FT     48000 FT  145 DEG
                    TURN LEFT       110 DEG    59534 FT   34 DEG
                    PROCEED      200000 FT    259534 FT   34 DEG

     ***************************   W A R N I N G   **************************
     A/C GONE BEYOND ALTITUDE PROFILE AFTER  3 STEPS
     ************************************************************************
                                                                  79.719 SEC

***  B-52G          AIRCRAFT NO. =      51     MISSION NO. =        1
                    OPERATIONS - DAY    3.020 , NIGHT     0.000
                       3000 FT RESTRICTION RESCINDED
```

```
------------------------------------------------------------------------

PROFILE GENERATED -- DISTANCE   ALTITUDE
                          0          0 FT
                       5000          0 FT
                      12000        100 FT
                      33306       3000 FT
                      48000       3000 FT
                      62694       5000 FT
                     214694      20500 FT

FLIGHT TRACK  --  MANEUVER             DISTANCE      HEADING
                  EXECUTED               AFTER COMPLETION
              PROCEED       48000 FT     48000 FT  145 DEG
              TURN LEFT       110 DEG    59534 FT   34 DEG
              PROCEED      200000 FT    259534 FT   34 DEG

***************************   W A R N I N G   ***************************
A/C GONE BEYOND ALTITUDE PROFILE AFTER  3 STEPS
*************************************************************************
                                                                70.639 SEC

***  LANDINGS      FLIGHT TRACK  14E
                   PROCEED     200000 FT

***  KC-135A       AIRCRAFT NO. =    251    MISSION NO. =      10
                   OPERATIONS - DAY    8.400 , NIGHT    -0.000        5.916 SEC

 **  B-52G         AIRCRAFT NO. =     31    MISSION NO. =      10
                   OPERATIONS - DAY    8.400 , NIGHT    -0.000        5.649 SEC

***  DUMP GRID TO UNIT 12
     PROGRAM CATALOGED FILE AND      SAVED DUMP   1 ON UNIT 12

***  DUMP GRID TO UNIT 15 PRINTABLE
     PROGRAM CATALOGED FILE AND      SAVED DUMP   1 ON UNIT 15 SPECIAL FORMAT

***  CLEAR GRID *** GRID CLEARED ***


                              --------------------------
***                               R U N W A Y    32
                              --------------------------
                   LENGTH  11829 FT, GLIDE SLOPE  2.50 DEG, HEADING 325 DEG
                   START( 1644320,  664260)  END( 1638610,  674620)
                   DISPLACEMENTS - TAKEOFF     -0, LANDING      -0

***  DEPARTURE PROCEDURE

               CLIMB TO           3000 FT    THEN
               TURN TO HEADING     065
```

```
          PROCEED FOR      200000 FT

          **********  END OF PROCEDURE **********


***  KC-135A       AIRCRAFT NO. =    251    MISSION NO. =      2
                   OPERATIONS - DAY   21.610 , NIGHT    0.000

     PROFILE GENERATED -- DISTANCE   ALTITUDE
                                0          0 FT
                             6000          0 FT
                            10000        200 FT
                            18000       1000 FT
                            27000       1000 FT
                           200000      18660 FT

     FLIGHT TRACK  --  MANEUVER             DISTANCE      HEADING
                       EXECUTED              AFTER COMPLETION
               PROCEED       46592 FT        46592 FT   325 DEG
               TURN RIGHT      101 DEG       57155 FT    65 DEG
               PROCEED      200000 FT       257155 FT    65 DEG

     **************************     W A R N I N G     **************************
     A/C GONE BEYOND ALTITUDE PROFILE AFTER  3 STEPS
     ****************************************************************************
                                                                    63.009 SEC

 **  DUMP GRID TO UNIT 15 PRINTABLE  SAVED DUMP   2 ON UNIT 15 SPECIAL FORMAT

***  DUMP GRID TO UNIT 12            SAVED DUMP   2 ON UNIT 12

***  CLEAR GRID *** GRID CLEARED ***


***  EXPUNGE FLIGHT DESCRIPTORS
                             AC CLASS  MISSION  OPERATION  DESCRIPTOR
                             --------  -------  ---------  ----------
                                 1        1      TAKOFF    * NOT FOUND *
                                51        1      TAKOFF      B-52G
                               251        2      TAKOFF      KC-135A
                               251        3      TAKOFF    * NOT FOUND *

***  EXPUNGE FLIGHT DESCRIPTORS
                             AC CLASS  MISSION  OPERATION  DESCRIPTOR
                             --------  -------  ---------  ----------
                                 1       10      LANDNG    * NOT FOUND *
                                51       10      LANDNG      B-52G
                               251       10      LANDNG      KC-135A
                                 1       11      LANDNG    * NOT FOUND *
```

```
**   EXPUNGE ALTITUDE PROFILES
           NAME
         --------
            11      * NOT FOUND *
           511
          2512
          2513
            51
            52      * NOT FOUND *

***  EXPUNGE POWER LEVEL PROFILES
           NAME
         --------
            11      * NOT FOUND *
           511
          2512
          2513
            60      * NOT FOUND *
            61
            62

***  FLIGHT DESCRIPTOR  AIRCRAFT - B-52G B2       TURN RAD -     6000.0FT
                       OPERATION - TAKOFF         ALT PROF -     5122
                        CLASS NO -      51        POW PROF -     5122
                      MISSION NO -      22
                       SUBFLIGHT - 1     NOISE PROF -      511
                                       TRACK LIMITS -        0 TO   18000 FT
                       SUBFLIGHT - 2     NOISE PROF -      512
                                       TRACK LIMITS -    18000 TO 1000000 FT

***  FLIGHT DESCRIPTOR  AIRCRAFT - K135A B3       TURN RAD -     6000.0FT
                       OPERATION - TAKOFF         ALT PROF -    25123
                        CLASS NO -     251        POW PROF -    25123
                      MISSION NO -      23
                       SUBFLIGHT - 1     NOISE PROF -     2511
                                       TRACK LIMITS -        0 TO   18500 FT
                       SUBFLIGHT - 2     NOISE PROF -     2512
                                       TRACK LIMITS -    18500 TO 1000000 FT
```

```
------------------------------------------------------------------------------

**   ALTITUDE PROFILE          NAME =     5122
         TRACK DIST      ALTITUDE
       ------------   ------------
                0 FT          0 FT
             6000 FT          0 FT
            10500 FT        330 FT
            18000 FT       1200 FT
            74700 FT       1200 FT
           101700 FT          0 FT

***  POWER LEVEL PROFILE       NAME =     5122
         TRACK DIST    REL POWER (DB)
       ------------   -------------
                0 FT         0.0
            18000 FT         0.0
            18500 FT         1.3
            74200 FT         1.3
            74700 FT        -1.0
           101700 FT        -1.0

***  ALTITUDE PROFILE          NAME =    25123
         TRACK DIST      ALTITUDE
       ------------   ------------
                0 FT          0 FT
             4500 FT          0 FT
             8500 FT        200 FT
            18500 FT       1200 FT
            80700 FT       1200 FT
           107700 FT          0 FT

***  POWER LEVEL PROFILE       NAME =    25123
         TRACK DIST    REL POWER (DB)
       ------------   -------------
                0 FT         0.0
            18500 FT         0.0
            19000 FT         7.5
            80200 FT         7.5
            80700 FT         2.9
           107700 FT         2.9


                                   -------------------------
***                                   R U N W A Y    14
                                   -------------------------
                      LENGTH  11829 FT, GLIDE SLOPE  2.50 DEG, HEADING 145 DEG
                      START( 1638610,  674620)  END( 1644320,  664260)
                      DISPLACEMENTS - TAKEOFF   1500, LANDING      0

***  TAKE-OFFS        FLIGHT TRACK  1482
                      PROCEED       14000 FT
                      TURN LEFT     180 DEG   WITH    6000 FT   RADIUS
                      PROCEED       32000 FT
                      TURN LEFT     180 DEG   WITH    6000 FT   RADIUS
                      PROCEED       18000 FT
```

** B-52G B2 AIRCRAFT NO. = 51 MISSION NO. = 22
OPERATIONS - DAY 25.200 , NIGHT -0.000 75.608 SEC

*** TAKE-OFFS FLIGHT TRACK 1483
PROCEED 17000 FT
TURN LEFT 180 DEG WITH 6000 FT RADIUS
PROCEED 35000 FT
TURN LEFT 180 DEG WITH 6000 FT RADIUS
PROCEED 18000 FT

*** KC135A B3 AIRCRAFT NO. = 251 MISSION NO. = 23
OPERATIONS - DAY 25.200 , NIGHT -0.000 85.374 SEC

*** DUMP GRID TO UNIT 12 SAVED DUMP 3 ON UNIT 12

*** DUMP GRID TO UNIT 15 PRINTABLE SAVED DUMP 3 ON UNIT 15 SPECIAL FORMAT

*** CLEAR GRID *** GRID CLEARED ***

*** RUNUP NOISE LEVEL PROFILE (PNLT) NAME = 535

	ANGLE IN DEGREES								
DIST	0.0	10.0	40.0	60.0	120.0	130.0	140.0	150.0	180.0
200.0 FT	131.2	132.0	132.3	132.9	142.1	143.0	139.8	129.8	109.8
250.0 FT	129.1	129.8	130.1	130.7	139.9	140.9	137.6	127.7	107.7
315.0 FT	126.8	127.6	127.8	128.5	137.7	138.7	135.5	125.5	105.5
400.0 FT	124.5	125.3	125.6	126.2	135.5	136.5	133.3	123.4	103.4
500.0 FT	122.3	122.9	123.2	123.8	133.3	134.3	131.1	121.2	101.2
630.0 FT	119.9	120.4	120.7	121.3	130.9	131.9	128.7	118.8	98.8
800.0 FT	117.5	118.0	118.1	118.7	128.4	129.5	126.3	116.5	96.5
1000.0 FT	114.9	115.5	115.3	115.9	125.8	126.9	123.8	114.0	94.0
1250.0 FT	112.3	112.8	112.5	113.0	123.1	124.2	121.1	111.3	91.3
1600.0 FT	109.5	110.0	109.6	109.9	120.2	122.0	118.3	108.6	88.6
2000.0 FT	106.5	107.0	106.6	106.7	117.8	119.1	115.9	106.1	86.1
2500.0 FT	104.5	105.0	103.8	103.3	114.0	115.3	112.2	102.0	82.0
3150.0 FT	100.7	101.2	99.7	99.7	109.3	111.2	107.6	98.2	78.2
4000.0 FT	96.6	96.8	95.6	95.5	105.0	106.8	103.2	95.0	75.0
5000.0 FT	92.0	92.5	92.3	92.1	101.5	102.9	99.8	90.7	70.7
6300.0 FT	87.4	86.4	86.2	86.0	96.9	96.8	93.8	84.6	64.6
8000.0 FT	83.6	84.0	82.1	81.9	91.5	93.1	90.1	80.8	60.8
10000.0 FT	79.5	79.9	77.7	77.5	87.2	90.3	86.2	76.9	56.9
12500.0 FT	75.1	75.5	72.9	72.7	82.9	86.2	81.9	72.7	52.7
16000.0 FT	70.3	70.7	69.2	67.4	79.2	81.7	77.4	68.1	48.1
20000.0 FT	64.8	65.2	63.7	61.9	73.1	76.8	73.5	64.0	44.0
25000.0 FT	59.3	59.7	58.2	56.4	67.0	71.9	69.6	59.9	39.9

*** RUNUP DESCRIPTOR

AC CLASS	THRUST	DESCRIPTOR	PNLT PROF	PNLT OFFSET
251	100	KC-135A	535	-8.00

```
--------------------------------------------------------------------------------

**  RUNUP DESCRIPTOR
      AC CLASS    THRUST    DESCRIPTOR  PNLT PROF  PNLT OFFSET
      --------   --------  ----------  ---------  -----------
         51        100      B-52G         535        -8.00


                         --------------------------------
***                       R U N U P    P A D  1K135-1
                         --------------------------------
     X =  1639060 FT, Y =   669520 FT, HEADING = 276.0 DEG.
                                      RUNUPS PER TIME PERIOD  DURATION OF
        AIRCRAFT CLASS     THRUST   0700-2200  2200-0700    EACH RUNUP
***       251 KC-135A        100       2.600       0.000    120.000 SEC


                         --------------------------------
***                       R U N U P    P A D  3B52-4
                         --------------------------------
     X =  1637470 FT, Y =   672380 FT, HEADING = 234.0 DEG.
                                      RUNUPS PER TIME PERIOD  DURATION OF
        AIRCRAFT CLASS     THRUST   0700-2200  2200-0700    EACH RUNUP
***        51 B-52G          100       4.350       0.000    100.000 SEC

***  DUMP GRID TO UNIT 15 PRINTABLE  SAVED DUMP   4 ON UNIT 15 SPECIAL FORMAT

***  DUMP GRID TO UNIT 12            SAVED DUMP   4 ON UNIT 12

***  ADD DUMP    1 FROM UNIT 12    SKIPPED    0 DUMPS AFTER REWIND
     DUMP WAS WRITTEN AS DUMP    1 ON UNIT 12 BY PROGRAM NEFUSAF ON 12/20/73
     FROM AIRFIELD COMPUTER PROGRAM OPERATOR'S MANUAL EXAMPLE


***  ADD DUMP    2 FROM UNIT 12    SKIPPED    0 DUMPS
     DUMP WAS WRITTEN AS DUMP    2 ON UNIT 12 BY PROGRAM NEFUSAF ON 12/20/73
     FROM AIRFIELD COMPUTER PROGRAM OPERATOR'S MANUAL EXAMPLE


***  ADD DUMP    3 FROM UNIT 12    SKIPPED    0 DUMPS
     DUMP WAS WRITTEN AS DUMP    3 ON UNIT 12 BY PROGRAM NEFUSAF ON 12/20/73
     FROM AIRFIELD COMPUTER PROGRAM OPERATOR'S MANUAL EXAMPLE


***  DUMP GRID TO UNIT 12            SAVED DUMP   5 ON UNIT 12
                                  SKIPPED    1 DUMPS

***  DUMP GRID TO UNIT 15 PRINTABLE  SAVED DUMP   5 ON UNIT 15 SPECIAL FORMAT
```

```
***********************************************************************

***  ENTER NON-PROCESSING MODE
     INPUT DATA WILL BE CHECKED, BUT NO CONTOUR COMPUTATIONS PERFORMED

***********************************************************************

***  PLOT TRACKS    WIDTH 28 IN, OPT = 5, SCALE 1 TO  50000 (1 IN =   4167 FT)
                    GPCP CONTROL CARDS ON UNIT 11 TRANSFERRED FROM UNIT  8
                    CONTOURS ARE    --    30   40
                    THERE ARE   25 ANNOTATION RECORDS      0 NEF DATA POINTS

***********************************************************************

***  ENTER PROCESSING MODE
     CONTOUR COMPUTATIONS WILL BE PERFORMED

***********************************************************************

***  PLOT CONTOURS WIDTH 28 IN, OPT = 5, SCALE 1 TO  50000 (1 IN =   4167 FT)
                    GPCP CONTROL CARDS ON UNIT 11
                    CONTOURS ARE    --    30   40

                    RUNWAY LAYOUT SUPPRESSED  6800 NEF DATA POINTS
```

```
ERROR STATISTICS
================

          FATAL ERRORS =      NONE


          WARNING MESSAGES =  OCCUR ON PAGE(S)
                                  6    7    8
```

BBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBN

BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN

U N I T E D S T A T E S A I R F O R C E

NOISE EXPOSURE FORECAST

12/20/73

TERMINATION PROCEDURE

BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBN UNITED STATES AIR FORCE NOISE EXPOSURE FORECAST COMPUTER PROGRAM BBN
BBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBNBBN

```
FILES REFERENCED DURING THIS RUN

LOGICAL UNIT 10 CONTAINS   0 DUMPS(PHYSICAL)
LOGICAL UNIT 15 CONTAINS   5 FORMATTED DUMPS
LOGICAL UNIT 12 CONTAINS   5 DUMPS(PHYSICAL)   5 (LOGICAL)

THE OPCP INTERFACE HAS BEEN ENDFILED ON UNIT 11
```

APPENDIX B

ERRORS, WARNINGS AND MESSAGES

This appendix contains a listing of the errors, warnings and messages discussed in the body of the report except that those messages which constitute an echo of a data card are omitted.

The program is capable of issuing more messages than these but they are not further discussed in the text. This is the set of self explanatory messages such as

A '*******' CARD PRECEDES THE FIRST 'AIRFLD' CARD

RUNUP DESCRIPTOR CONTINUATION CARD NOT FOUND

The messages are listed in alphabetical order. When the value of a variable is part of a message this value is represented by the star symbol (*) whether this value is numeric or alphabetic. The star symbol has no value in the collating sequence of the messages.

A PREVIOUS ENTRY FOR **** HAS BEEN DELETED	93
A/C GONE BEYOND ALTITUDE PROFILE AFTER ** STEPS	137
A/C IS ALREADY AT ALT. **** FT WHEN **** FT INSTRUCTION IS GIVEN	140
A/C TURN RADIUS *** TO LARGE FOR INTERCEPT OF *** *** RADIAL. A ** DEGREE COURSE ADJUSTMENT IS MADE	140
AIRCRAFT IS NOT AIRBORNE AT THE START OF TURN	113
AIRCRAFT IS STILL ON THE GROUND AT THE END OF THE RUNWAY	113
AIRCRAFT NUMBER SPECIFIED ON THE FLIGHT CARD IS NOT PRESENT IN THE DIRECTORY	113
AIRCRAFT REMAINS ON GROUND DURING FIRST INSTRUCTION	129
ALL CONTOURS REQUESTED WERE SUPPRESSED	164
ALTITUDE **** CANNOT BE REACHED	139
ALTITUDE OR DELTA-EPNL PROFILE UNDEFINED	73
ALTITUDE PROFILE NO. ***** SPECIFIED IN FLIGHT DESCRIPTOR HAS NOT BEEN ENTERED	113
ALT. REST. **** FT ILLEGAL AFTER **** FT WHEN A/C IS AT **** FT	123, 139

ANGLES NOT IN ASCENDING ORDER OR DUPLICATE ANGLE	85
ANGLE SPECIFIED ON FLTTRK CARD IS GREATER THAN 360 DEGREES	110, 111
CANNOT PROCEED DIRECT TO ** NAVAID'DIST = **** FT RADIUS = **** FT A/C CONTINUES ON HEADING *** DEG	141
CONTOURS BELOW 25 ARE NOT CONSIDERED RELIABLE	163
CONTOURS BELOW 25 ARE NOT CONSIDERED RELIABLE FURTHERMORE CONTOURS BELOW 15 ARE SUPPRESSED	164
CURRENT GRID STATUS INACCESSIBLE SAVED DUMP *** ON UNIT 10	31
DISTANCE COVERED BY SUBFLIGHTS IS LESS THAN THE TOTAL FLIGHT TRACK	113
END CARD ENCOUNTERED DURING INITIALIZATION	40
ENTRY **** NOT KNOWN	94
ESSENTIAL ALTITUDE **** CANNOT BE REACHED	139
ESSENTIAL NAVAID MISSING	138
ESSENTIAL *** *** RAD ALREADY PAST	138
ESSENTIAL *** *** RAD NOT INTERSECTED	138

EXECUTION OF PROCEDURE SKIPPED DUE TO PREVIOUS ERROR(S)	122
EXECUTION TERMINATED DUE TO INITIALIZATION ERRORS	101
FILE CATALOG FULL	16, 32
FLIGHT TRACK DOES NOT START WITH A LINE SEGMENT	109, 111
FLIGHT TRACK TURN RADIUS < ******.* FT AS SPECIFIED ON 'FLTDSC' CARD	74
*** FORCED ENTRY	95
GPCP CONTROL CARDS ON UNIT **	163
GPCP CONTROL CARDS ON UNIT ** TRANSFERRED FROM UNIT **	163
ILLEGAL AC/MISSION NOS	73
ILLEGAL CONTINUATION AFTER 'FLTTRK' CARD	112
ILLEGAL ENTRY **** ****	128
ILLEGAL ENTRY TURN ****	128
ILLEGAL ENTRY TURN *** AS FIRST INSTRUCTION	128
ILLEGAL GLIDE SLOPE	106
ILLEGAL MAGNETIC DECLINATION ***.* DEG TO ****	103

ILLEGAL NAME	53, 60, 66, 81
ILLEGAL RESTRICTION -- ALTITUDE = **** FT DISTANCE = **** FT NAVAID = *** *** RAD	132
ILLEGAL TAPE HEADER	18, 36
INITIAL TRACK DIST NOT ZERO	62, 68
INTEGRATED NOISE LEVEL PROFILE NO. ***** SPECIFIED IN FLIGHT DESCRIPTOR HAS NOT BEEN ENTERED	113
INVALID (as an aircraft type)	112, 118
INVALID AC CLASS, THRUST, OR PNLT PROF	88
INVALID NAME AND/OR PROPAGATION CODE NAME = ***** P.C. = *****	55
INVALID UNITS SPECIFICATION - EXECUTION TERMINATED	42
LANDING DISPLACEMENT IS ILLEGAL	106
MANIPULATION WITH ILLEGAL FILE	33
MAX. NOISE LEV. PROF. **** MISSING	118
MISSING CONTINUATION CARD	55, 76

MISSING CONTINUATION CODE OR MISSING DATA	56
MISSING CONTINUATION CODE OR MISSING DATA. LAST ANGLE = ***.*	85
MISSING DATA. LAST ANGLE = ***.*	84
MODE CONFLICT UNIT IS BINARY .. DUMPED TO UNIT 6	32
MODE CONFLICT UNIT IS PRINTFILE .. DUMPED TO UNIT 10 SAVED DUMP *** ON UNIT 10	32
NAME DOES NOT MATCH FOR ANGLE = *****	83
NAVAID NAME MISSING	93
NAVAID *** NOT KNOWN	137
NAVAID MISSING *** (IT IS IGNORED)	137, 138
NOISE LEVEL DATA OUT OF RANGE	56
NOISE LEVELS DO NOT DECREASE FOR ANGLE = ***.* DEG	84
NOISE LEVELS NON-DECREASING FOR PRPGTN CODE = **	56
NOISE LEVEL(S) OUT OF RANGE	84
NO KNOWN DUMPS ON THIS UNIT	34

NOT FOUND	57, 63, 69, 78, 86, 91
NUMBER OF COORDINATES RESTRICTED 2 TO 10	62, 68
NUMBER OF SUBFLIGHTS RESTRICTED 1 TO 3	76
OFFSET OUT OF RANGE	90
ONLY *** DUMPS ON UNIT **	35
OUT OF RANGE	56, 68
PAD IN USE MORE THAN 15 HOURS/DAY	118
PAD IN USE MORE THAN 9 HRS/DAY	118
PAPER SIZE TOO SMALL, SIZE LEFT AT **.* INCH	43
PLOT OPTION ******* IS ILLEGAL; A VALUE OF 2 IS ASSUMED	165
POWER LEVEL PROFILE NO. ***** SPECIFIED IN FLIGHT DESCRIPTOR HAS NOT BEEN ENTERED	113
PROCEDURE ABANDONED AFTER ** STEP(S)	123, 172
PROCEDURE TOO COMPLEX	142
PROCEDURE TOO COMPLEX ALTITUDES	142

PROCEDURE TOO COMPLEX DISTANCES	142
PROCEDURE TOO COMPLEX HEADINGS	142
PROCEDURE TOO COMPLEX NAVAIDS	142
PROFILE GENERATED EXCEEDS STORAGE AVAILABLE	141
PROGRAM CANNOT READ PRINTFILE BACK	32
PROGRAM CATALOGED FILE AND SAVED DUMP 1 ON UNIT ***	16, 32
*** *** RAD DELETED FROM RESTRICTION	134
*** *** RAD DELETED FROM RESTRICTION *** FT RESTRICTION RESCINDED	134
*** *** RAD NOT INTERSECTED	134
REL POWER OUT OF RANGE	68
RESTART FAILED - NOT ALL ERRORS WERE CORRECTED	175
RESTRICTION PRECEDES FIRST INSTRUCTION	124
RESTRICTIVE ALTITUDE *** NOT FOUND IN STEP * (MAX ALT = *** FT).	139
RUNUP DESCR. FOR THIS COMBIN. MISSING	118

RUNWAY LAYOUT SUPPRESSED **** NEF DATA POINTS	165
RUNWAY LENGTH IS GREATER THAN 16000 FT	105
SCALE 1 TO ******* ILLEGAL, SET TO 1 TO 24000	165
SUBFLIGHT END DIST MUST BE GREATER THAN BEGIN DIST	76
TABLE FULL	49, 93
TAKEOFF DISPLACEMENT IS ILLEGAL	106
THERE ARE **** ANNOTATION RECORDS ***** NEF DATA POINTS	164
TOO MANY ANGLES, 10 MAX	86
TOO MANY SEGMENTS IN FLIGHT PATH	111
TOO MANY SEGMENTS IN FLIGHT TRACK	110, 111
TRACK DISTANCE(S) NOT POSITIVE OR NOT ASCENDING	62, 68
UNIT ** BLOCKED (LIMITED ACCESS DUE TO ILLEGAL HEADER)	175
UNIT ** CONTAINS *** LOGICAL BUT ONLY *** PHYSICAL DUMPS SAVED DUMP *** ON UNIT 10	34, 169
UNIT ** INPUT ONLY (LIMITED ACCESS DUE TO ILLEGAL HEADER)	175

```
UNIT ** NOW WRITE ACCESSIBLE.
                                                                        36

WEIGHTED OPERATIONS ***.*** ILLEGAL NO COMPUTATION FOR THIS FLIGHT      113
```


☆U.S.Government Printing Office: 1974 — 657-013/47